電腦時代 COMPUTING AGE 13

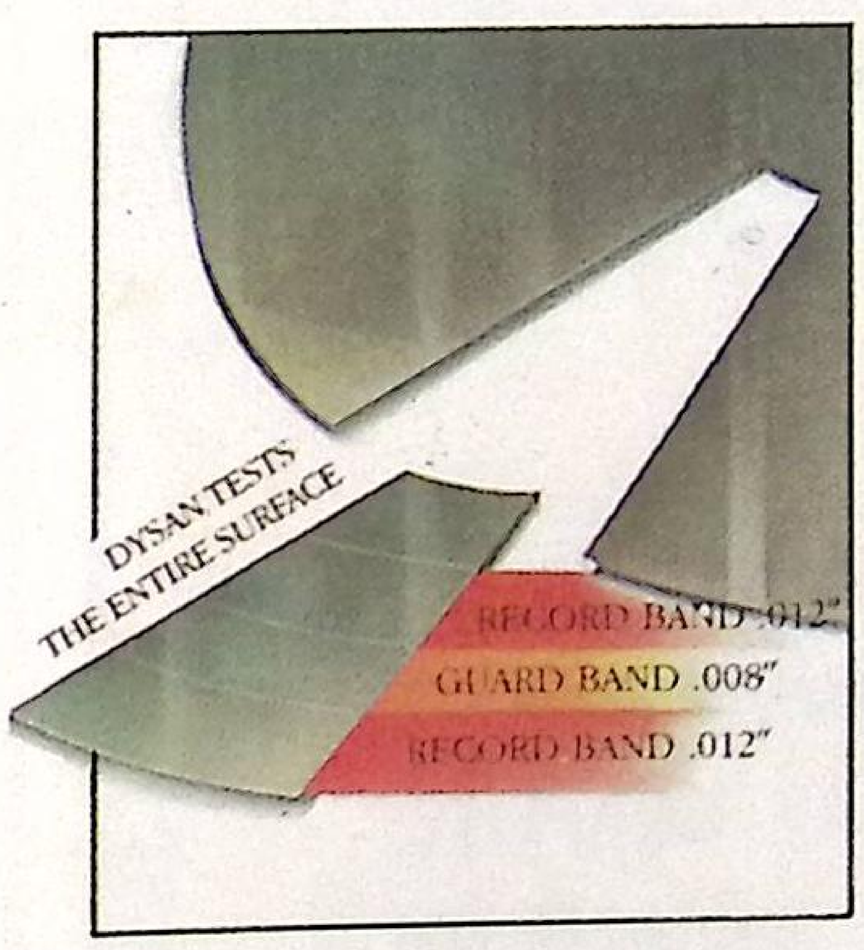
微型電腦用家雜誌 HK$8.00

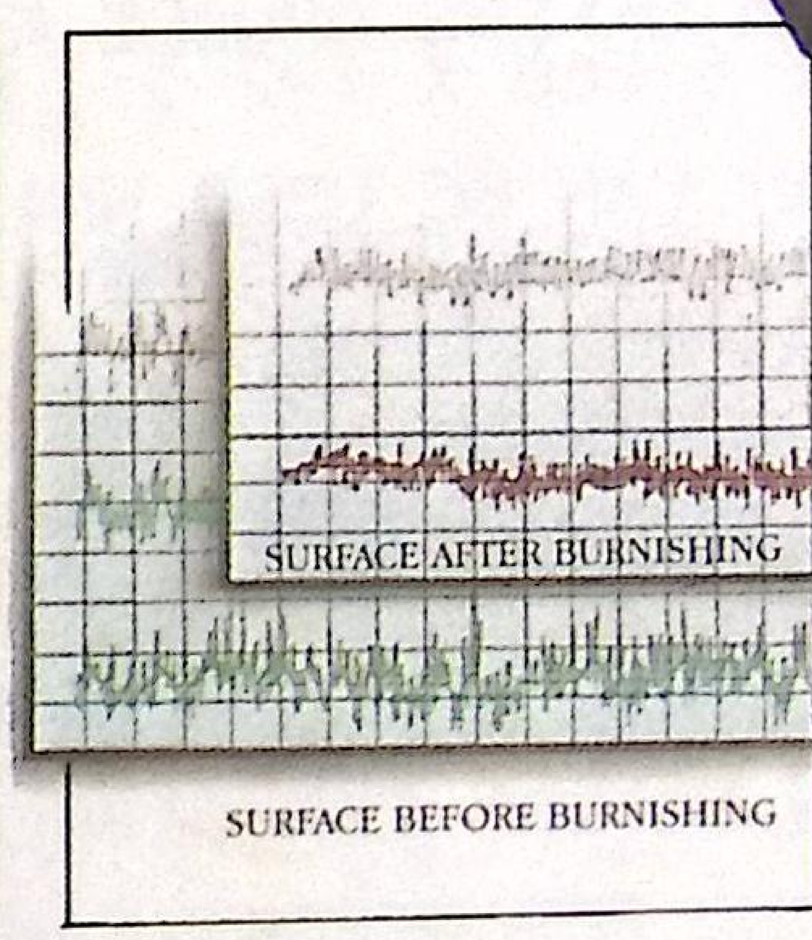

TAXAN RGB 「幻彩系列」
彩色電腦螢幕
色彩艷麗完美無瑕
● 能顯示任何顏色組合
率先採用「線性影像放大系統」
● 絕不干擾磁碟及其他電器
設計符合多項美國嚴格標準
● 適用於各款電腦
備有多種附件，適用於下列電腦：
RGB（紅綠藍）彩色介面
APPLE II¹ APPLE IIe¹（連80字顯示）
介面接駁線
APPLE III¹ IBM PC² BBC³ NEC PC⁴
FUJITSU FM-7⁵ COMMODORE⁶
歡迎訂造特別介面接駁線
1. APPLE IS A TRADE MARK OF APPLE COMPUTOR INC
2. IBM IS A TRADE MARK OF INTERNATIONAL BUSINESS MACHINE CORP
3. BBC IS A TRADE MARK OF ACORN INTERNATIONAL LTD.
4. NEC PC IS A TRADE MARK OF NEC CORP.
5. FUJITSU FM-7 IS A TRADE MARK OF FUJITSU LTD.
6. COMMODORE IS A TRADE MARK OF COMMODORE ELECTRONICS LTD.
RGB VISION-III
TAXAN
享有代理商保證
才是專業性保用
總代理
VINCENT COMPUTER CENTRE LTD.
(A Subsidiary of Vincent Group)
雲生電腦有限公司
HX38316 VCC HK

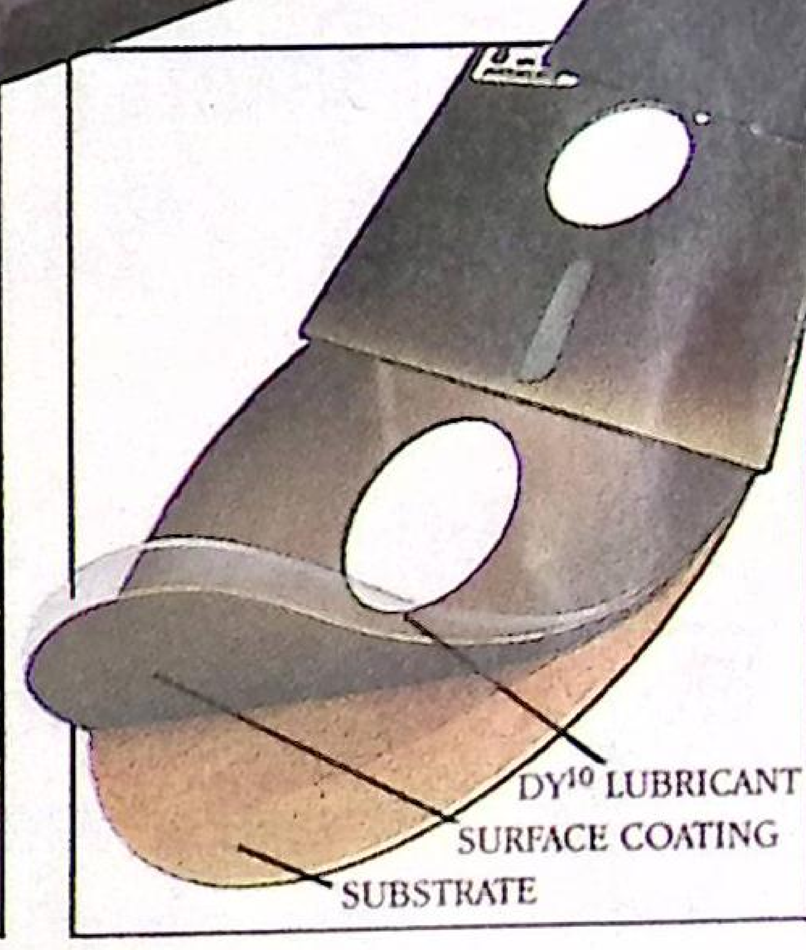
DISCOVER THE DYSAN DIFFERENCE

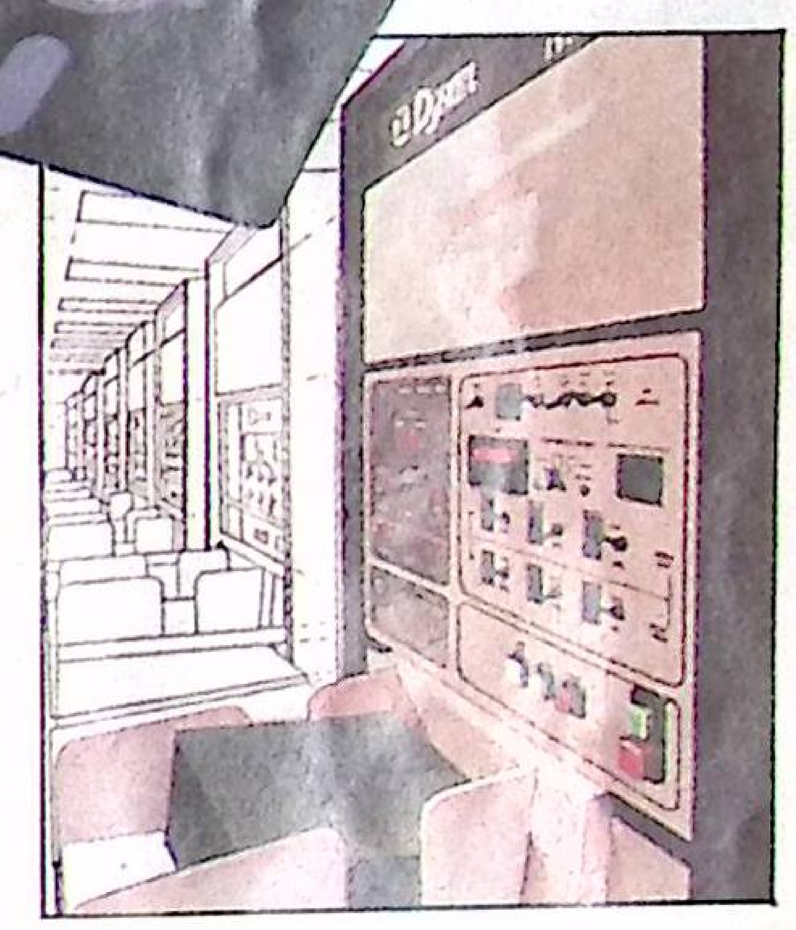
Four Reasons Why The Dysan Difference is Worth Paying For











1. 100% Surface Tested

Only Dysan provides fully usable diskette surfaces that are truly 100% error-free across the entire face of the diskette. An exclusive on-and-between the track testing procedure guarantees error-free performance regardless of temperature and humidity distortions or slight head misalignments.

2. Advanced Burnishing Techniques

Dysan's advanced polishing methods create a smoother, more uniform diskette surface. This results in better signal quality on each track, less wear on drive heads and reliable access to data after millions of head passes.

3. DY^{10}™ Lubricant

Dysan's proprietary DY^{10} lubricant complements the advanced burnishing process. Both maximize error-free performance while minimizing headwear. Optimal signal presence is maintained between the head and diskette surface during millions of write/read interfaces.

DY^{10} is a trademark of Dysan Corporation

4. Auto-Load Certification

Dysan's unique quality control methods reflect technological leadership in designing, producing and testing precision magnetic media. Each diskette is unerringly certified by Dysan-built, automated and microprocessor controlled certifiers. Your system and data base will benefit from Dysan's diskette reliability and unsurpassed quality.

Authorized Distributor:

Supply Consultants Ltd.
ROOM 1802 VALLEY CENTRE, 18/F.,
80-82 MORRISON HILL ROAD, HONG KONG.
TELEPHONE: 5-8917697 TELEX: 61627 SUPAV HX

Dysan® CORPORATION

第一期內容

陽曆陰曆日子對換表

一個輸入西曆日子便可查到相對的農曆日子工具。農曆日子以天干地支來表示。

音樂火花一龍的傳人

作者利用豐富的高解像畫面襯托出一首美妙動人的樂章

五骰遊戲

可以兩人玩或和電腦做對手的擲骰遊戲，除了要靠彩數外，還要憑個智慧取勝的決擇。

記憶能力的考驗

需要高度集中精神來玩的一個遊戲，玩者要憑記憶力來找到一對對的搭子，有多種難度供升級之用。

物理學的氣壓定律

特別為中學四、五年級的同學而編寫的一個實用程式，讓同學們能溫習所學習的氣壓定律。

電子電話部

這個電話部的特點是在檢索人名地址時，非常之快，而且使用十分容易，一用即會。

波子機彈子原理模擬

讀者未透過這個程式從中獲得設計彈子反彈反應的原理，在編寫有關的程式時，可收事半功倍之效。

快速抄碟程式 COPY ＋

使用這個拷貝工具，單用戶磁碟機的讀者只要換三次碟即可將一隻沒有保護的程式拷貝至一份。

將 COPY A 及 DISK ZAP 改抄 40 軌磁碟

四十軌磁碟將會大行其道，一張四十軌的磁碟較３５軌的多出１５％的記憶空間可用。但COPY A 或DISK ZAP 本身是不能進行４０軌的拷貝，本文教你如何改動程式的來適應需求。

CONTROL RESET 使用方法

RESET 鍵是蘋果鍵盤中最特別的一個鍵，在使用上，大家或許未能充份掌握，本文作者利用多個實例來闡釋這個鍵的用途。

魔水晶二號破解方法

魔水晶二號的原作者共花了７２步來尋求魔水晶，但本港一個電腦迷，比作者行少１步便找到魔水晶，証明本港實是有一流高手的存在。

美國最新出版軟件榜

大家必然對蘋果軟件的發展很關心。這個資料榜將近五個月的蘋果軟件名單蒐集一起，讓大家了解及認知。

電子郵箱

為讀者而設的一個閱讀信箱，讀者可利用此信箱來信詢問各類有關的問題。我們定當盡力以最快的速度來處理詢問

數碼式電子鐘

無名的遊戲

一行程式集

商品公司廣告集

封面高解像版頭設計

尚有其他……

第二期十二月底出版

內容包括：

陽曆陰曆日子對算表資料輸入法

創刊號中登載的〔西曆查農曆日子〕只能查閱１９２４年至１９８６年之間的日子，以後那些年份便沒有效。事實本程式並沒有年份限制，今期我們教你如何輸入１９８７年以後年份的資料來查閱。

磁碟雜誌中文字顯示原理剖析（一）

磁碟雜誌的內容說明主要是用中文，但又不用中文咭來輔助，究竟為何能做到這種效果，本文將會由淺入深的為你剖析。

化學科模擬選擇題練習

為了讓中五的同學溫習過去所學的，以應備會考，我們由今期開始為大家提供各科的會考模擬試題練習，首先是化學科。

接連火車卡模擬

這是一個有趣的拼字遊戲模擬，利用一輛火車頭在迷宮中穿梭將亂放的英文字拉併成完整的一個英文字，過程非常有趣。

高登最受歡迎遊戲精選

十二月份高登中最受歡迎的一個遊戲軟件，我們選出來供大家娛樂。

撲克遊戲之一排七

排七是紙牌遊戲中最受歡迎的一種。我們將它寫成電腦程序來玩，遊戲的難度及趣味性大大的提高。

支持老鼠仔的軟件介紹

坊間雖已有老鼠出售，價錢不算貴，但目前本港有關支持老鼠仔的軟件很少，我們蒐集了現時美國所有支持老鼠仔的軟件供各位參考。

SUPPER HELLO

這是一個有趣的HELLO程式，在CATALOG 時，它會自動將各類型的檔案分類，使一目了然。

俄羅斯輪盤

大家不用去葡京，反之安坐家中便可玩輪盤遊戲。這裡最公平不過，你可試試自己的運氣。

一九八四年全美四十大最受歡迎遊戲軟件

數據處理庫

尚有其他……

COMPUSOFT 磁碟月刊
Disk Magazine
Vol. 1 No. 1 November, 1984.

親愛的讀者們：

你們好！COMPUSOFT磁碟月刊創刊號和你們見面了。希望你們喜歡它，支持它。

電腦時代出版社

香港蘋果電腦用家踏入歷史性新里程

COMPUSOFT 磁碟月刊

每期出版兩張$5\frac{1}{4}$"雙面灌錄的磁碟（合共４面）

內容包括：

1. 多個極富娛樂刺激的電腦遊戲
2. 不同取材的文章，適合任何行業人士閱讀
3. 提高學能的教育軟件（由小學到中學），教師最宜參考
4. 題材廣泛的電腦知識專題講座，提高學習電腦能力
5. 難以衡量價值的實用工具程式，幫助使用電腦
6. 最新軟件（遊戲／商業軟件）測試報告，了解最新軟件
7. 繞樑三日的音樂軟件，百聽不厭
8. 最新最熱門軟件BACK UP方法／參數
9. 最新出版中、英文版本電腦書籍消息，毋需逛書局
10. 最新出版商業、娛樂軟件簡介，掌握市場動態
11. 最新出產電腦產品介紹
12. 各大電腦公司產品上市消息
13. 讀者園地／信箱　歡迎一切的投稿、詢問
14. 每期封面設計比賽
15. 還有其他許多許多難以在此盡錄……

COMPUSOFT 磁碟月刊是以DOS 3.3 FORMAT，適用於任何蘋果或蘋果相容的電腦。閱讀本刊，毋需專門電腦知識，只要略懂簡單的電腦基本操作方法即行。每期內容有詳細的目錄索引，閱讀時非常方便。磁碟內容除有英文詳細說明外，**亦附有中文清楚解說，讀者絕對不需要使用中文咭便可顯示中文解說**，閱讀時絕對沒有語言的阻礙。

訂閱COMPUSOFT磁碟月刊辦法簡單，訂戶每期收到「磁碟」後，自行COPY 一份，然後寄回磁碟給我們，我們將會在下期繼續寄出。當然，讀者亦可前來電腦時代出版社讀者服務部攞取每期磁碟月刊。

要令你或你的家人得益，今日就請立即訂閱〔COMPUSOFT磁碟月刊〕。

訂閱COMPUSOFT 磁碟月刊，每期只需15元。COMPUSOFT 磁碟月刊 只接受訂閱

讀者如有任何問題可電熱線電話3-322110查詢。

只適用於本港訂戶

本人欲訂閱 COMPUSOFT磁碟月刊

- ☐ 12期　HK $180（退回磁碟）
- ☐ 12期　HK $540（連同磁碟）*
- ☐ 6期　HK $100（退回磁碟）
- ☐ 6期　HK $280（連同磁碟）*
- ☐ 1期試閱　HK $ 50（連同磁碟）*

由第____期開始

支票收款人請填寫 "Computing Age Publisher"

請寄九龍中央郵箱71193號

信封面請註明「訂閱」

姓名：__________

____

地址：__________

____

____

電話：__________

附上劃線支票 No. __________

（請用英文填寫，方便電腦處理）

NO.__________
Official Use

重要啟事

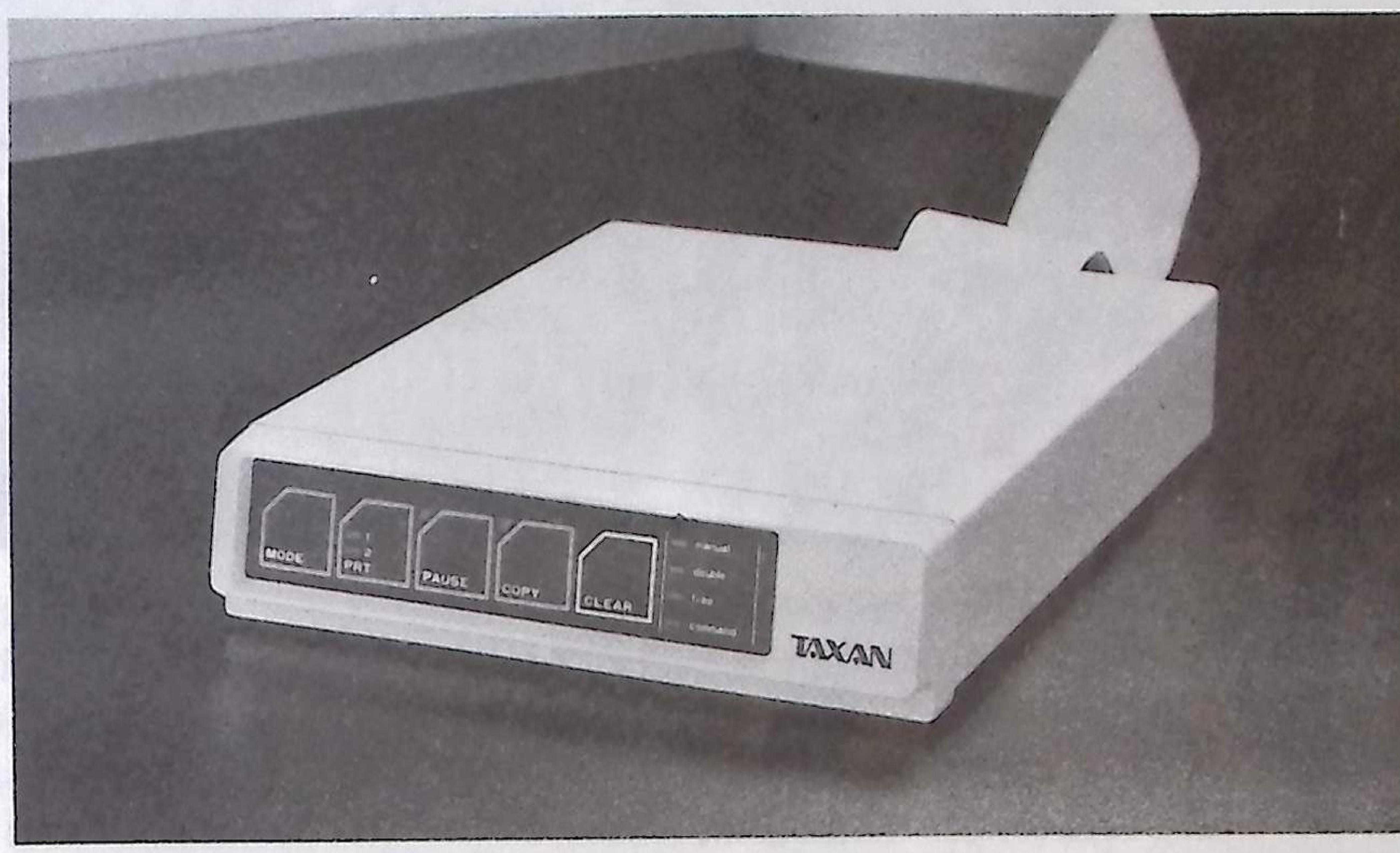
本公司自1979年成立至今，並未有設立任何分公司。近日市面上，有一電腦公司因慕名而冒名使用與本公司相似之名稱，售買「假貨」及訛稱由本公司負責維修保養。

本公司一向不售賣任何「假貨」及無其它分號，故此請各消費者加倍小心，免被漁目混珠，招至損失。

本公司並不排除對該等不法商人採取一切法律行動。敬請認明本公司之英文全名串法及爲有限公司。

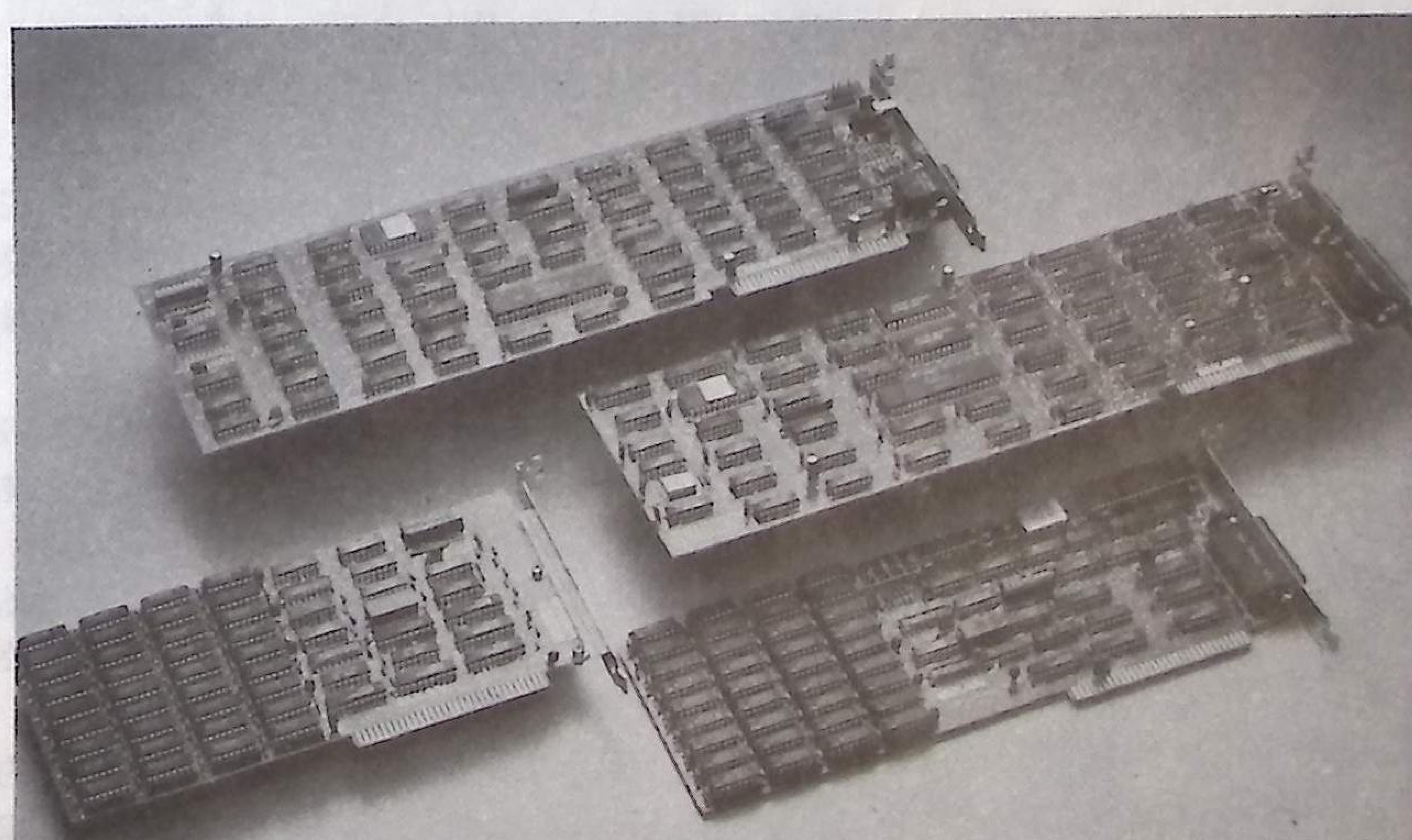
TAXAN®

週邊介面



KIF 4308印字機記憶緩衝器

- 64 K 記憶系統可另增加記憶至256 K。
- 不佔用電腦時間，尤其在印中文字及圖案時。
- 可作無數次COPY。增加工作效率。
- 同時可供兩部印字機使用，可由硬件或軟件控制。



IBM PC*及PC/XT*專用介面

- KIF 3210：256K RAM記憶擴展介面。
- KIF 3210 S：256 K RAM記憶擴展介面及串行式介面接口。
- KIF 3600：彩色／圖案顯示介面。
- KIF 3700：單色顯示及CENTRONICS平行式印字機介面。

總代理：

VINCENT COMPUTER CENTRE LTD.

雲生電腦有限公司

九龍尖沙咀柯士甸路22-26號好兆年行608室 電話：3-7211288-0 電訊：HX38316 VCC HK

*IBM PC,PC/XT are registered trademark of International Business Machine Corp.



近日高登商場出售的一張新週邊咭MOCKINGBOARD，相信不少讀者都略有所聞。這張咭令到APPLE電腦的發聲效果大大的提高。爲了讓大家能深入的認識這張咭，筆者花盡心思搜集了許多有關資料介紹給大家。

MOCKINGBOARD 的製造商是美國的SWEET MICRO SYSTEMS（地址是：150 CHESTNUT STREET PROVI—DENCE，RI 02903）。這張咭可任意揷在各個SLOT上，除了SLOT 0外。

SWEET MICRO SYSTEM 推出的這張咭其實有四種版本供選擇。

(1) SOUND Ⅰ
(2) SOUND Ⅱ
(3) SPEECH Ⅰ
(4) SOUND／SPEECH Ⅰ

SOUND Ⅰ純粹是一張聲響效果產生器。SOUND Ⅱ是雙重音響效果產生器；SPEECH Ⅰ是一張語音合成器；而SOUND／SPEECH Ⅰ是音響效果及語音合成器的結合體。各個版本中，以SOUND／SPEECH Ⅰ 售價最貴，在美國約售170 美元。所有四張咭都是採用同一晶片設計及發展技術，所以基本上沒有大分別。

MOCKINGBOARD 的使用手册詳述了每種版本的使用方法。

MOCKINGBOARD 的能產生奇妙聲響效果全賴GEN—ERAL INSTRUMENT的AY—3—8910 PROGRAMMA—BLE SOUND GENERATOR（PSG）。這枚晶片提供有三聲道的音調，因而能產生豐富的音響效果。每聲道都有獨立的音調控制。當然這枚晶片更重要一項特點是它只使用非常少的CPU時間來INITIALIZE 聲音的啓始，之後它就不會再用CPU的時間來維持發聲操作。換言之，設計程式時，可以任意在高速動畫中配上音響效果，而不會影響螢幕畫面。

SPEECH BOARD使用VOTRAX SC—01 語音合成器。這枚晶片能產生64個音素（PHONEME ，即英文中的基本語音），換言之，這枚語音合成器能製造任何的字彙。

使用MOCKINGBOARD 時，必須要另外配合一對揚聲器，或者接連到擴音器上。這塊咭包括有一個½W的放大器來推動一部8Ω的揚聲器。在音量方面，足夠令你有如置身在電子遊戲機中心裏。假如你嫌音量過大，可調校在咭頂上的音量控制器，每聲道都有獨立的音量控制器。此外，MO—CKINGBOARD 亦附有調頻器來改變音質。

假如是採用雙重音响效果的版本，如SOUND／SPEECH Ⅰ，必須要用兩具揚聲器。留意一件事情，如果你嘗試經“Y”接連器驅動兩聲道輸出同一揚聲器，則會產生失眞的效果。

MOCKINGBOARD 的使用手册，並不如咭本身的一樣受人讚嘆。全書只用上三十餘頁的篇幅便概括了所有的版本及其使用方法。由於這咭並不支持BASIC 解釋器，要製造聲音及音响就必須用機械語言（編者按：請參閱同期麥法基先生的MOCKINGBOARD一文）來存取16個不同的暫存器位置。SWEET MICRO 提供有一張磁碟來錄載一些基本的組合語言編寫的現成可用音响效果。不過只會 BASIC 的朋友亦不用太担心不會使用這些子程式，手册上的解釋會令你滿意。

SWEET MICRO SYSTEM 同時提供有一張TEXT TO SPEECH ALGORITHM（由文字轉爲語音）的磁碟，這張磁碟包括有一個SOUND EDITOR，一個PHONEME EDITOR，及多個的預定聲响效果。這些程式可以讓一個完全不懂組合語言的電腦用家來製造聲音及音响效果。透過這些程式，用家只要打入一段文字，便可將它轉化爲人類的對話。

要製造他人明白的語音，必須要掌握程式中一些基本的音素規則或者透過EDITOR 來自己定出規劃。（編者註：我們將於下期登載一個供MOCKINGBOARD使用的SPEECH EDITOR，大家請留意）■

以MOCKING BOARD 來製造音響效果的技巧

麥法基

蘋果電腦無疑是一個多才多藝的電腦，無論商業，教育及遊戲都有大批軟件支持。我們現在可以擁有各式各樣的動畫遊戲，ADVENTURE 等。但總是欠缺一些東西——完善的聲音系統，倘若APPLE 能供應各種不同音响效果和包括不同和絃的音樂，不是更完美得多嗎？

最近黃金商場地庫有一張MOCKINGBOARD 發售，這張介面卡是針對聲音而設計的。安裝後APPLE 會變成一個可以產生立體音响效果的電腦，對於使用APPLE 的人會覺得進入另外一個境界。我們可以得到更迫眞的音响例如海浪衝擊聲，由遠而近的火車或脚步聲，震天般的爆炸聲及其他效果。進一步我們亦可以創造六度和絃的美妙音樂，所有這些聲音都是身歷其境的立體聲，APPLE 再不是一個欠缺聲音的電腦了。

MOCKINGBOARD 是一張自己擁有擴音部份的介面卡，只要接上一對喇叭便可以調節聲音的大小，卡上更有兩個空置的I.C.插座用以擴展成爲一張說話卡，把卡插在APPLE 的第 4 SLOT 上便可以應用了。

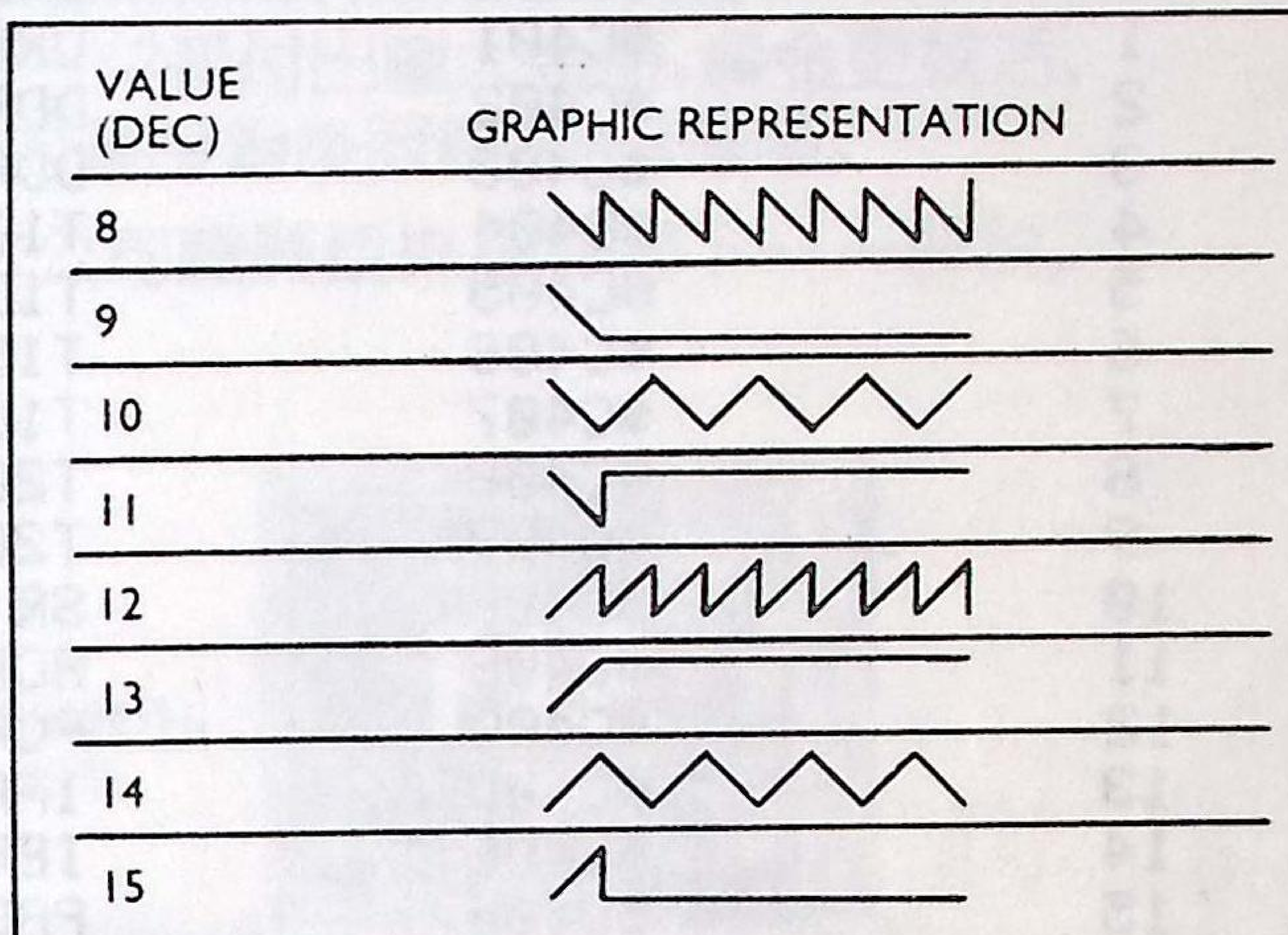
MOCKINGBOARD 的最大功用是給程式設計者加以運用在程式內，所以隨卡附有一張SOUND UTILITY 的磁碟。好像設計動畫程式一樣，聲音亦同時需要一個SOUND TABLE 。不同的聲音需要不同SOUND TABLE的資料，而磁碟上的SOUND UTILITY 程式是給程式設計者預先創造聲音表，自己滿意後便可以加入程式內。筆者在這裏並不打算介紹如何把SOUND TABLE 加入程式內。有興趣的朋友請參考同期刊出的POPYE 遊戲的SOURCE LISTING。不過亦可以簡略介紹一下SOUND UTILITY 的用法。

從圖一我們知道聲音包括各種不同的ADJUSTMENTS，調節分四大部份進行。第一部份是 TONE／NOISE PERIOD：TONE PERIOD 是純音樂調節。其中包括COARSE 和FINE 兩種調節方法，CORSE 是粗略的調節，由 0 數值開始的最尖音調，數值越大則音調越沉，通常我們可以接受美妙悅耳的音樂大約由 0 至 7 總共可以產生八個八度音程的樂音，而大部份的音樂都在這個範圍以內，通常以鋼琴爲例都大約七個八度音程而已。以音樂而言，每個音程都有不同的音符，電腦當然可以產生更細緻的頻率，就像小提琴的絃綫一樣，這就需要倚靠FINE 的調節來達到目的了。COARSE和FINE 的關係就像時鐘的分針和秒針一樣互相調節而產生準確的效果。NOISE PERIOD 是噪音的調節，數值由 0 至31，數值小是噪音比較尖銳而數值大則噪音沉重，舉例而言就像用尺輕輕敲打的聲音和用鎚子大力敲打的分別。不過有很多時候我們不需要純音和噪音混合以產生不同的效果。

一、

SPEAKER: 1

REGISTER NAME	MAX VAL	ALL	CHANNEL A	B	C
TONE PER FINE	255	0	0	0	0
TONE PER COARSE	15	0	5	5	5
NOISE PERIOD	31	10			
ENABLE	63	0			
AMPLITUDE (FIX=0-15/VAR=16)	16	0	10	10	10
ENVL PER FINE	255	0			
ENVL PER COARSE	255	0			
ENVL SHAPE	15	0			



二、

第二部份是ENABLE ，由於MOCKINGBOARD 每邊喇叭可以同時產生三個純音和三個噪音；這是很重要的調節。因爲未必每次同時產生六種聲音，而混合噪音和樂音亦由這調節控制。

第三部份是音量調節AMPLITUDE ，由 0 至15是連續性的16個不同大小音量。第十七個是根據下面第四部份的音形而產生聲音。

第四部份ENVELOPE／ENVELOPE SHAPE 音形調節。每種聲音均有一種特殊的PATTERN 加以循環。音形通常有不同音量組合而成，而ENVELOPE 均有COARSE 和FINE 等兩種細緻的調節。ENVL SHAPE供應八種不同的音形，選擇如圖二。例如我們希望得到強大的爆炸聲音，可以選擇第 9 種音形，聲音使會經過壓縮後突然產生爆炸聲，然後慢慢沉寂下來。倘若改爲第10種音形，則會變成爆炸後隨著有海浪衝擊的波浪聲不絕於耳。

當我們調節了自己滿意的音响效果後，便可以按 S 把音响表SOUND TABLE 存放到磁碟上加以運用了。

MOCKINGBOARD 產生聲音並不會使我們的程式減慢

以MOCKING BOARD 來製造音響效果的技巧

三、

Register	Address	Symbol	Name
0	$C400	ORB	I/O Port B
1	$C401	ORA	I/O Port A
2	$C402	DDRB	Port B data register
3	$C403	DDRA	Port A data register
4	$C404	T1CL	Timer 1,low order
5	$C405	T1CH	Timer 1,High order
6	$C406	T1LL	low-order latch,T1
7	$C407	T1LH	high-order latch,T1
8	$C408	T2CL	Timer 2,low order
9	$C409	T2CH	Timer 2,high order
10	$C40A	SR	Shift Register
11	$C40B	ACR	Auxilary Control
12	$C40C	PCR	Peripheral Control
13	$C40D	IFR	Interrupt Flag
14	$C40E	IER	Interrupt Enable
15	$C40F	PAD	Same as 1

速度，它可以說是一種獨立處理聲音的媒介。當獲取SOU—ND TABLE 後，產生的聲音均會獨立處理，因爲它是以兩枚6522的ＩＣ作爲輸入／輸出介面。我們並不需要另外再學習6522的程式設計，因爲6522本身只是一個VIA （VERSATIVE INTERFACE ADAPTER)，它是設計6502 的人員另外創立的一種純爲6502 作爲輸入／輸出及時間控制的集成電路。所以我們可以稱之爲6502的FAMILY-TYPE集成電路，它可以十分容易利用6502相同的BUS ADRESS和SIGNAL 作爲介面處理，所以只要同樣以6502的程式設計便可以產生兩個輸入／輸出。當MOCKINGBOARD 插在APPLE的第４SLOT時，C400和C401便構成6522的兩個輸入／輸出埠了。原則上我們可以把MOCKINGBOARD插在第一至第七個SLOT 上加以運用，不過MOCKING—BOARD 的內在程式以第４SLOT 爲主，而大部份程式設計亦以爲４SLOT 作爲與6522的溝通，所以通常都把它插在第４SLOT上。參看圖三和圖四便可以知道MOCKING—BOARD 把所有6522的輸入／輸出地址作爲聲音的媒介了。

獨立處理聲音使程式設計者可以很容易加入各種音响，背景音樂，讓各種遊戲或教育程式更加生色不少。況且不用照顧聲音時間問題，程式運行自然加快了，不用担心程式會加長，因爲MOCKINGBOARD 只佔用少量的記憶。而且一MANNUAL 裏附有發聲程式的原始碼，方便程式設計者把程式改變或存放在任何地址上加以運用。磁碟上亦有SOU—ND LIBRARY 存着很多不同的音响表供選擇應用，並且有現成的程式可供參考之用。由於所有聲音決定於音表SO—UND TABLE，只要改變SOUND TABLE 的資料便可以改變不同的聲音，這對於程式設計者來說，是十分輕而易舉的事。

四、

	REGISTER	DESCRIPTION	DEC	HEX
R0	CHANNEL A TONE PERIOD	FINE TUNE	0-255	00-FF
R1		COARSE TUNE	0-15	00-0F
R2	CHANNEL B TONE PERIOD	FINE TUNE	0-255	00-FF
R3		COARSE TUNE	0-15	00-0F
R4	CHANNEL C TONE PERIOD	FINE TUNE	0-255	00-FF
R5		COARSE TUNE	0-15	00-0F
R6	NOISE PERIOD	ALL CHANNELS	0-31	00-1F
R7	ENABLE	NOISE/TONE	0-63	00-3F
R8	CHANNEL A AMPLITUDE	AMPLITUDE LEVEL MODE SELECT FIXED = 0-15 VARIABLE = 16	0-16	00-10
R9	CHANNEL B AMPLITUDE		0-16	00-10
R10	CHANNEL C AMPLITUDE		0-16	00-10
R11	ENVELOPE PERIOD	FINE TUNE ENVELOPE	0-255	00-FF
R12		COARSE TUNE ENVELOPE	0-255	00-FF
R13	ENVELOPE SHAPE/CYCLE	CONT:ATTACK:ALT: HOLD	0-15	00-0F

R14 NOT USED REGISTER VALUE NOT SIGNIFICANT
R15 NOT USED REGISTER VALUE NOT SIGNIFICANT

如果想知道MOCKINGBOARD 吸引的地方，把MUSIC CONSTRUCTION SET 這程式LOAD入便立刻可以欣賞到很多首美妙的樂曲了。■

最精明方式選購蘋果電腦

精明人比其他人更懂得享受金錢的樂趣。正如閣下現在選購蘋果電腦，就可以利用太平洋行「月供電腦計劃」，以極易負担的金額，享受先進精密的蘋果電腦威力。

閣下現時到本公司以分期付款方式選購蘋果IIe、蘋果IIc或Macintosh™，每月只需供款低至$538起，不但毋須一次過付出龐大金額，資金更靈活，並且可享受蘋果特約分銷商才能提供的優良銷後服務，更有保障。

有關太平洋行「月供電腦計劃」詳情，本公司職員極樂意爲閣下一一解說，請即日駕臨參觀試用。

蘋果IIe 月供$551

用途多樣化，爲全球最受歡迎之私人電腦。除主機外，並包括微型磁碟驅動器，螢幕顯示器等，足以處理日常各項計算事宜。





蘋果IIc 月供$538

蘋果電腦最新產品，袖珍式體積，重量級威力。記憶容量達128K，內置微型磁碟驅動器，新型鍵盤，9吋傾斜螢幕顯示器及六款示範軟件。



蘋果Macintosh™ 月供$825

Macintosh™ 爲當今電腦科技中最尖端之設計。主體爲32位元微型處理器，內置三吋半微型磁碟驅動器及革命性「老鼠仔」設計。

Authorized Dealer

特約分銷商：

EXCELLENT ELECTRONIC CO. LTD. 第一(潘氏)電腦有限公司

九龍油麻地彌敦道343-349號(大華戲院對面)永安置業大厦1301室 電話：3-313878 3-323563

發聲的鍵盤

令你的APPLE有大機風範

亞崇

筆者日前突然興起寫PROGRAM 的興趣，於是叠埋心水，星期日留在AP Ⅱ旁KEY IN PROGRAM 。但越撳越覺得AP Ⅱ的鍵盤不對勁，正所謂毫無聲氣，一些觸感也沒有。要是能像IBM或其它MICRO機般在每撳一個鍵時便發出一聲〝嗶〞聲的話，KEY IN PROGRAM的樂趣定可大大提高。

於是翻查REFERENCE MANUAL ，發覺AP Ⅱ的KEYBOARD CONNECTOR 的第二脚是鍵盤的STROBE輸出。在每撳一下鍵時，它便輸出一個PULSE 。

本來理論上我們是可以直接利用這些PULSE 來控制一個音頻產生綫路，以達到上述的目的，但可惜每個PULSE只有十多個μS 。以這麽短的時間來控制音頻綫路，不論是喇叭或人耳也是來不及作出反應的。

解決的方法是加一個延遲電路（TIME DELAY CIRCU—IT) 以增加PULSE 的寬度，然後才以這些PULSE 控制音頻電路。

圖一是這個〝KEYBOARD BEEPER〞的全綫路圖。

R1，R2，R3及Q1 組成一個簡單的反相輸入緩衝器。

IC1，R4，C2 等組成一個延遲電路，反相了的STRO—BE PULSE 由IC1的第二脚輸入，而第三脚則輸出已擴寬的PULSE ，輸出PULSE 的寬度由R4及C2的數值所控制，改變R4或C2的數值能改變發聲的時間長短。

IC2，Q3，R6，R7，C3 等組成一個音頻產生器，這個產生器的頻率則受R6，R7，C3的值控制。

而Q2，R5 則是一個簡單的開關電路，當IC1 第三脚的輸出是〝1〞時，Q2 便導通而使喇叭發聲。當IC1 第三脚的輸出是〝0〞時，Q2 則保持開路，阻止音頻信號到達喇叭。

整個綫路非常簡單，可以直接銲在一塊萬用底板上。而綫路中的喇叭更可省去而把Q2的C 脚直接接在AP Ⅱ 內置喇叭的信號脚上，但切要小心不可接錯，由爲AP Ⅱ內置喇叭的另一隻脚是接在＋5 V電源的。爲安全計，還是另外接一個喇叭好，反正一個小型的喇叭也只是三兩塊錢而已。

經過試驗後，發覺整個綫路工作得非常好，音量也算適中。閣下只要花費十多元，便能使閣下心愛的AP Ⅱ的KEY—BOARD〝生聲〞起來，實在是物有所值。

筆者不才，如有錯漏，還望各方高人不吝賜教，萬分感激。■

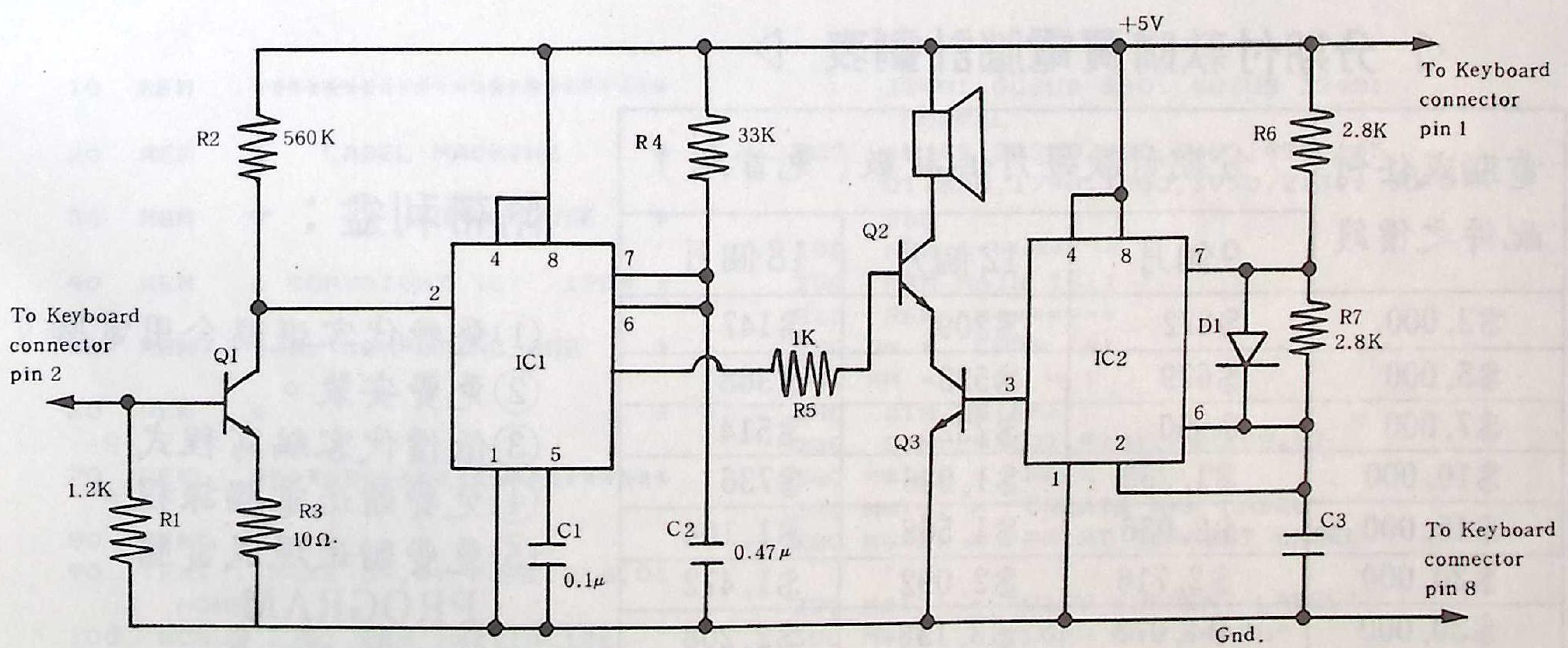


圖1

IC1，IC2 ：NE555×2

Q1，Q2，Q3：BC108×3

D1：IN4001

+5V	1	16	NC
STROBE	2	15	−12V
RESET	3	14	NC
NC	4	13	Do
D5	5	12	D1
D4	6	11	D3
D6	7	10	D2
Gnd	8	9	NC



學習尖端科技
邁向成功坦途

現代電腦訓練中心
香港電腦推廣協會 合辦

全港首創—分期付款購買電腦辦法

(可購買IBM, EAGLE, EVERGO, PC-88, APPLE-IIe, COLUMBIA及APPLE-II與IBM之相容機種。亦適合購買上述電腦之MONITOR, DRIVER, PRINTER等。)

◁ 分期付款購買電腦計劃表 ▷

電腦或任何配件之價錢	分期付款每月供款數(免首期)		
	9個月	12個月	18個月
$2,000	$272	$209	$147
$5,000	$679	$523	$368
$7,000	$950	$732	$514
$10,000	$1,358	$1,046	$736
$15,000	$2,036	$1,568	$1,104
$20,000	$2,716	$2,092	$1,472
$30,000	$4,072	$3,136	$2,208
$50,000	$6,788	$5,228	$3,680

附帶利益：-

①免費代客選購合用電腦。
②免費安裝。
③低價代客編寫程式。
④免費贈送電腦課程。
⑤免費贈送現成電腦PROGRAM
⑥一年免費保養。

地址：九龍深水埗福華街156號黃金商場地庫·20號及54號

電話：3-601531, 3-869085

本程式較諸現有的MAILING程式使用 更方便

每個標貼最多可印十行及每行32個字

全能標貼製作器

成明

藉着這個全能標貼製作器，閣下可以製作最多至一百個的標貼，存放在一張磁碟上，並隨自己喜愛的格式編製出來。對於有許多郵寄地址名單或各種說明用途貼標的用家來說，本程式是最適合不過，有了這個全能標貼製作器，各位就不用再受那些MAILING LIST PROGRAM 的死板格式所限制了。

全能標貼製作器能在一個標貼上印出最多十行，每行32個英文字母的內容，較過去任何的MAILING LIST PROG—RAM 都方便。

除了這項優點外，製作器是十分容易使用。大家只是用左右箭咀鍵來作選擇，並用RETURN鍵來執行。

當執行程式後，螢幕會出現一個主索引，包括九項：

(1) 製作新標貼
(2) 印出記憶體中的標貼
(3) 觀看記憶體中的標貼
(4) 載入舊的標貼
(5) 存入記憶體中標貼入磁碟
(6) 刪除舊標貼
(7) 觀看舊標貼
(8) 將標貼分類
(9) 寫入檔案及離開程式

本程式會將每個標貼的第一行作為該標貼的記錄，大家在使用時留意。

在編印標貼時，各位可用放大字體、正常字體、或濃縮字體、重行字體。當各位選擇第二項功能時，製作器自會詢問你。按SPACE 表示放棄該種字體，而按RETURN 則表示使用這種字體。

這個程式所配用的印字機是ESPON MX80 或相容的印字機。各位朋友可能會使用不同的印字機，爲此，我編寫了一個輔助選擇印字機副程式，（檔案名字叫SET PRINTER）帮助各位以最簡單的方法來設定匹配的印字機。■

```
10   REM  ************************
20   REM  *   LABEL MACHINE      *
30   REM  *    BY  COMPUHOUSE    *
40   REM  * COPYRIGHT (C)  1984 *
50   REM  * BY COMPUTING AGE     *
60   REM  *                      *
70   REM  ************************
80   REM
90   TEXT : POKE 34,0: POKE 216,0:
      HOME
100  GOSUB 220: REM INITIALIZE
110  GOSUB 410: REM TITLE
120  GOSUB 2960: REM CHECK FOR FI
     LE
130  ONERR  GOTO 150
140  VTAB 23: HTAB 9: PRINT "PRES
     S ANY KEY TO START": GOSUB 3
     190: GOTO 160
150  CALL  - 3288
160  GOSUB 500
170 HB = 1: HOME : INVERSE : GOSUB
     3240: GOSUB 660: GOSUB 3240:
      NORMAL
180  ON SE GOSUB 700,860,1490,155
     0,1640,1790,1880,1930,2130: GOTO
     160
190  REM =========
200  REM MAIN INIT
210  REM =========
220 D$ =  CHR$ (4)
230 AM = 9:CL = 1
240  DIM M$(AM)
250  DIM L$(10),P(3),K$(100,9)
260 M$(0) = "MAIN MENU "
270 M$(1) = "CREATE NEW LABEL"
280 M$(2) = "PRINT CURRENT LABEL
     "
290 M$(3) = "VIEW CURRENT LABEL"
300 M$(4) = "LOAD OLD LABEL"
310 M$(5) = "SAVE CURRENT LABEL"
320 M$(6) = "DELETE OLD LABEL"
330 M$(7) = "VIEW OLD LABELS "
340 M$(8) = "SORT LABELS "
350 M$(9) = "WRITE FILE AND QUIT
     "
360 L$(0) = "....................
     ............": REM 32 DOTS
370  RETURN
380  REM =========
```

```
390  REM MAIN TITLE
400  REM ==========
410  VTAB 3: INVERSE : GOSUB 3240
420  PRINT "       PERFECT LABEL CR
     EATING  MACHINE       ";: GOSUB
     3240: REM  12 AND 11 SPACES
430  PRINT "           BY    COMP
     UHOUSE               ";: GOSUB
     3240: REM  11 AND 10 SPACES
440  PRINT "**  COPYRIGHT 1984 BY
      COMPUTING AGE  **";: GOSUB
     3240
450  NORMAL
460  RETURN
470  REM =========
480  REM MAIN MENU
490  REM =========
500  HOME :HB = 1:I = 0: INVERSE
     : GOSUB 3240: GOSUB 660: GOSUB
     3240: NORMAL
510 BB = 0: FOR I = 1 TO AM:B =  LEN
     (M$(I)): IF B > BB THEN BB =
     B
520  NEXT I
530 HB =  INT ((40 - (BB + 4)) /
     2) + 1
540 VB =  INT ((24 - (AM + 5)) /
     2) + 4
550  VTAB VB + AM + 1: HTAB 1: FOR
     I = 1 TO 40: PRINT  CHR$ (45
     );: NEXT
560  VTAB VB + AM + 3: PRINT "  <
     = => SELECT           <RTN>
     EXECUTE  "
570  FOR I = 1 TO AM: GOSUB 650: NEXT
     :I = 1: INVERSE : GOSUB 650:
      NORMAL
580  GOSUB 3190
590  IF X = 13 THEN SE = I: RETURN

600  IF X = 21 AND I < AM THEN  GOSUB
     650:I = I + 1: INVERSE : GOSUB
     650: NORMAL : GOTO 580
610  IF X = 8 AND I > 1 THEN  GOSUB
     650:I = I - 1: INVERSE : GOSUB
     650: NORMAL : GOTO 580
620  IF X = 21 AND I = AM THEN  GOSUB
     650:I = 1: INVERSE : GOSUB 6
     50: NORMAL : GOTO 580
630  IF X = 8 AND I = 1 THEN  GOSUB
     650:I = AM: INVERSE : GOSUB
     650: NORMAL : GOTO 580
640  GOTO 580
650  VTAB VB + I - 1: HTAB HB
660  PRINT  SPC( ((40 - 2 * HB -
      LEN (M$(I))) / 2) + 1);M$(I
     ); SPC( ((40 - 2 * HB -  LEN
     (M$(I))) / 2) + 1): RETURN
670  REM ================
680  REM CREATE NEW LABEL
690  REM ================
700  IF L$(1) <  > "" THEN  GOSUB
     2860
710  VTAB 10: HTAB 1
720  FOR I = 1 TO 9: PRINT "  "I"
      TO END INPUT"
740 VV = 9:C = 0
750 VV = VV + 1:C = C + 1: VTAB V
     V: HTAB 5: GOSUB 3110:L$(C) =
     A$: IF L$(C) = "." THEN L$(C
     ) = "": GOTO 780
760  IF C = 10 THEN C = 11: GOTO
     780
770  GOTO 750
780 C = C - 1: VTAB 10: HTAB 1: CALL
      - 958: GOSUB 3230
790  VTAB 20: FOR I = 1 TO 40: PRINT
      CHR$ (95);: NEXT
800  VTAB 23: HTAB 11: PRINT "IS
     THIS CORRECT ? ";: GOSUB 321
     0
810  IF X = 78 THEN  GOSUB 2910
820  RETURN
830  REM ===================
840  REM PRINT CURRENT LABEL
850  REM ===================
860  IF L$(1) = "" THEN  RETURN
870  VTAB 6: GOSUB 3230
880  VTAB 16: FOR II = 1 TO 40: PRINT
      CHR$ (95);: NEXT
890  VTAB 20: HTAB 11: PRINT "PRI
     NT THIS LABEL ? ";: GOSUB 32
     10
900  IF X = 78 THEN  RETURN
910  VTAB 18: HTAB 1: CALL  - 958

920  PRINT "ENHANCED CHARACTERS ?
     ": PRINT "ENLARGED CHARACTER
     S ?": PRINT "CONDENSED CHARA
     CTERS ?"
930  VTAB 23: PRINT "<SPACE> TO F
     LIP         <RTN> TO ACCEPT"

940  FOR V = 18 TO 20: VTAB V: HTAB
     26: GOSUB 1450: NEXT
950  FOR V = 18 TO 20
960  VTAB V: HTAB 35: GET A$: IF
     A$ <  >  CHR$ (32) AND A$ <
      >  CHR$ (13) THEN 960
970  IF A$ =  CHR$ (13) THEN P(V -
     17) = 1: GOTO 990
980  VTAB V: HTAB 26: GOSUB 1440:
     P(V - 17) = 0
990  NEXT V
1000  VTAB 23: PRINT "
        ";: REM 15 SPACES
1010  VTAB 21: HTAB 1: CALL  - 95
     8: PRINT "START PRINTING IN
     COLUMN";: HTAB 28: PRINT CL;
     : HTAB 34: INPUT " ";A$: IF
     A$ = "" THEN 1040
1020 CL =  VAL (A$): IF CL < 1 OR
     CL > 70 THEN CL = 1: GOTO 10
     10
1030  VTAB 21: HTAB 28: CALL  - 8
     68: PRINT CL;
1040  VTAB 23: HTAB 1: CALL  - 86
     8: HTAB 11: PRINT "IS THIS C
```

各位業餘程序編寫員萬勿錯過本文

電腦時代爲你開闢一條新途徑

各位業餘程序編寫員，你們的軟件創作才華有沒有被人忽視？你們的心血作品有沒有人賞識？

我們知道，你們這群業餘程序編寫員絕大部份都是天才橫溢，所創作編寫的軟件並不遜色於任何的職業程序設計師，而且，有許多更是超乎市面要求的商業水準。你們的作品之中，有許多是經年累月的經驗。我們懇請你們不要埋沒自己的天才，將你的經驗創作成果與每一個電腦迷分享。

電腦時代出版社與 COMPUHOUSE 軟件製作所願意爲你全盤策劃出版你的軟件創作成果，讓它流傳到每個電腦用家手中。你絕對不用担心任何的問題，我們會代你効勞一切有關版權、出版、發行製作等問題，你只要安心在家創作便可。

我們擁有豐富經驗的行政人員、軟件工程師、編採出版人材。而且更具有一個廣大的市場發行網，我們的發行網除遍達香港外，還推廣到台灣、星馬泰等東南亞國家，而且更延伸至美國及加拿大。中國大陸這個龐大的市場當然也在我們足跡之下。

透過我們出版軟件，你將會得到最大的收穫。

我們龐大隊伍的生產綫，不單設備完善，且有足夠能力應付任何各方面的需求。

我們歡迎任何機種及類型的軟件創作，無論是ＩＢＭ、ＡＰＰＬＥ電腦、由娛樂軟件、教育應用至商業軟件，只要你們有信心認爲自己的作品有一定的商業水準，我們都樂意爲你出版。

歡迎你們携同作品前來本出版社商談，或電3—312007 與譚先生聯絡。

```
     ORRECT ? ";: GOSUB 3210
1050  IF X = 78 THEN 910
1060 CC$ = "": FOR SP = 1 TO CL -
     1:CC$ = CC$ + " ": NEXT : IF
     CL = 1 THEN CC$ = ""
1070 I$ =  CHR$ (9) + "80N": REM
     MAX CHAR. PER LINE
1080 T1$ =  CHR$ (27) +  CHR$ (69
     ): REM ENHANCED PRINT ON
1090 C1$ =  CHR$ (27) +  CHR$ (70
     ): REM ENHANCED PRINT OFF
1100 T2$ =  CHR$ (14): REM ENLARG
     ED PRINT ON
1110 C2$ =  CHR$ (20): REM ENLARG
     ED PRINT OFF
1120 T3$ =  CHR$ (15): REM CONDEN
     SED PRINT ON
1130 C3$ =  CHR$ (18): REM CONDEN
     SED PRINT OFF
1140  IF P(1) THEN G1$ = T1$
1150  IF  NOT P(1) THEN G1$ = ""
1160  IF P(2) THEN G2$ = T2$
1170  IF  NOT P(2) THEN G2$ = ""
1180  IF P(3) THEN G3$ = T3$
1190  IF  NOT P(3) THEN G3$ = ""
1200 GG$ = I$ + G1$ + G2$ + G3$
1210 G$ = C1$ + C2$ + C3$
1220  VTAB 18: HTAB 1: CALL  - 95
     8: PRINT
1230  PRINT "            TURN THE PR
     INTER ";: INVERSE : PRINT "O
     N";: NORMAL : PRINT ".": PRINT
     "   PRINTING WILL START IN C
     OLUMN ";CL;"."
1240  PRINT "     HIT <SPACE> TO
     START PRINTING,": PRINT "<ES
     C> TO ABORT AND RETURN TO MA
     IN MENU "
1250  GOSUB 3190: IF X <  > 32 AND
      X <  > 27 THEN 1250
1260  IF X = 27 THEN  RETURN
1270  FOR II = 10 TO 1 STEP  - 1
1280  IF L$(II) <  > "" THEN C =
     II: GOTO 1300
1290  NEXT II
1300  PRINT D$"PR# 1": FOR I = 1 TO
     C: PRINT GG$ + CC$ + L$(I) +
     G$: NEXT : PRINT D$"PR# 0"
1310  VTAB 18: HTAB 1: CALL  - 95
     8: PRINT
1320  PRINT "  HIT <SPACE> TO POS
     ITION NEXT LABEL,": PRINT : PRINT
     : PRINT "          <RTN> TO
     CONTINUE"
1330  VTAB 20: GOSUB 3190: IF X <
      > 32 AND X <  > 13 THEN 133
     0
1340  IF X = 13 THEN 1370
1350  PRINT D$"PR# 1": PRINT : PRINT
     D$"PR# 0"
1360  GOTO 1330
1370  PRINT : VTAB 18: HTAB 1: CALL
      - 958
1380  VTAB 20: HTAB 8: PRINT "PRI
     NT SAME LABEL AGAIN ? ";: GOSUB
     3210
1390  IF X = 78 THEN  RETURN
1400  PRINT "Y"
1410  VTAB 22: HTAB 8: PRINT "SAM
     E PRINTER SETTINGS ? ";: GOSUB
     3210
1420  IF X = 78 THEN 910
1430  GOTO 1220
1440  PRINT "(Y/";: INVERSE : PRINT
     "N";: NORMAL : PRINT ")";: RETURN

1450  PRINT "(";: INVERSE : PRINT
     "Y";: NORMAL : PRINT "/N)";:
     RETURN
1460  REM ==================
1470  REM VIEW CURRENT LABEL
1480  REM ==================
1490  IF L$(1) = "" THEN  RETURN

1500  VTAB 10: GOSUB 3230: VTAB 2
     3: PRINT "HIT ANY KEY TO RET
     URN TO MAIN MENU .....";: GOSUB
     3190
1510  RETURN
1520  REM ==============
1530  REM LOAD OLD LABEL
1540  REM ==============
1550  GOSUB 2600: GOSUB 2400
1560  VTAB 23: HTAB 11: PRINT "LO
     AD THIS LABEL ? ";: GOSUB 32
     10
1570  IF X = 78 THEN  RETURN
1580  IF L$(1) <  > "" THEN  GOSUB
     2860
1590  FOR II = 1 TO 10:L$(II) = K
     $(Q,II - 1): NEXT
1600  RETURN
1610  REM ==================
1620  REM SAVE CURRENT LABEL
1630  REM ==================
1640  IF L$(1) = "" THEN  RETURN

1650  VTAB 10: GOSUB 3230
1660  VTAB 23: HTAB 11: PRINT "SA
     VE THIS LABEL ? ";: GOSUB 32
     10
1670  IF X = 78 THEN  RETURN
1680  GOSUB 2600
1690 CH = 1
1700 N = N + 1: IF N > 99 THEN 17
     30
1710  FOR I = 1 TO 10:K$(N,I - 1)
      = L$(I): NEXT : HOME : VTAB
     12: HTAB 10: PRINT "CURRENT
     LABEL SAVED": VTAB 23: PRINT
     "HIT ANY KEY TO RETURN TO MA
     IN MENU.....";: GOSUB 3190
1720  RETURN
1730  VTAB 10: HTAB 1: CALL  - 95
     8: VTAB 15: HTAB 10
1740  INVERSE : PRINT "FILE IS FU
     LL " CHR$ (7) CHR$ (7): NORMAL
     : VTAB 23: PRINT "HIT ANY KE
     Y TO RETURN TO MAIN MENU ...
     ..";: GOSUB 3190
1750  RETURN
1760  REM ================
```

```
1770  REM DELETE OLD LABEL
1780  REM ================
1790  GOSUB 2600: GOSUB 2400
1800  VTAB 23: HTAB 10: PRINT "DE
     LETE THIS LABEL ? ";: GOSUB
     3210
1810  IF X = 78 THEN  RETURN
1820  FOR JJ = Q TO N: FOR II = 0
      TO 9:K$(JJ,II) = K$(JJ + 1,
     II): NEXT II: NEXT JJ
1830 N = N - 1:CH = 1
1840  RETURN
1850  REM ===============
1860  REM VIEW OLD LABELS
1870  REM ===============
1880  GOSUB 2600: GOSUB 2400
1890  RETURN
1900  REM ===========
1910  REM SORT LABELS
1920  REM ===========
1930  VTAB 15: HTAB 5: PRINT "THI
     S SORT WILL TAKE SOME TIME"
1940  PRINT : HTAB 12: PRINT "OK
     TO PROCEED ? ": GOSUB 3210
1950  IF X = 78 THEN  RETURN
1960  GOSUB 2600: IF N =  - 1 THEN
      RETURN
1970 CH = 1
1980  VTAB 8: HTAB 14: CALL  - 95
     8: PRINT "SORTING....."
1990  VTAB 10: POKE 34,10
2000 J =  - 1
2010 J = J + 1: IF J = N THEN  HTAB
     5: PRINT K$(J,0): GOTO 2060
2020 K = J
2030 K = K + 1: IF K > N THEN  HTAB
     5: PRINT K$(J,0): GOTO 2010
2040  IF  LEFT$ (K$(J,0),5) >  LEFT$
     (K$(K,0),5) THEN  FOR II = 0
      TO 9:TEMP$ = K$(J,II):K$(J,
     II) = K$(K,II):K$(K,II) = TE
     MP$: NEXT
2050  GOTO 2030
2060  POKE 34,0: VTAB 8: HTAB 1: CALL
      - 958
2070  VTAB 15: HTAB 11: PRINT J +
     1;" LABELS SORTED"
2080  VTAB 23: PRINT "HIT ANY KEY
      TO RETURN TO MAIN MENU.....
     ";: GOSUB 3190
2090  RETURN
2100  REM ====
2110  REM QUIT
2120  REM ====
2130  IF  NOT CH THEN 2310
2140  ONERR  GOTO 2320
2150  VTAB 15: HTAB 12: PRINT "PL
     EASE WAIT....."
2160  PRINT D$"OPEN LABEL.DATA"
2170  PRINT D$"CLOSE"
2180  PRINT D$"DELETE LABEL.DATA"
2190  PRINT D$"OPEN LABEL.DATA"
2200  PRINT D$"WRITE LABEL.DATA"
2210  PRINT N: IF N =  - 1 THEN 2
     300
2220  FOR J = 0 TO N
2230  FOR I = 0 TO 9
2240  FOR S = 1 TO  LEN (K$(J,I))
2250  PRINT  MID$ (K$(J,I),S,1);
2260  NEXT S
2270  PRINT  CHR$ (92)
2280  NEXT I
2290  NEXT J
2300  PRINT D$"CLOSE LABEL.DATA"
2310  POKE 216,0: HOME : GOSUB 41
     0: PRINT : PRINT : END
2320  CALL  - 3288: POKE 216,0: VTAB
     15: HTAB 12
2330  PRINT  CHR$ (7)" ***  ERROR
      *** " CHR$ (7): PRINT : PRINT
2340  PRINT "CHECK YOUR DISK AND
     PRESS <RETURN> TO": PRINT "T
     RY AGAIN."
2350  GET W$: PRINT
2360  VTAB 15: HTAB 1: CALL  - 95
     8: GOTO 2140
2370  REM ==============
2380  REM DISPLAY LABELS
2390  REM ==============
2400 R =  - 1
2410  VTAB 10: CALL  - 958: IF R =
     N THEN  POP : RETURN
2420 R = R + 1: HTAB 5: PRINT K$(
     R,0): IF R < N AND  VAL ( RIGHT$
     ( STR$ (R),1)) < 9 THEN 2420
2430  IF SE = 7 THEN 2450
2440  FOR S = 0 TO  VAL ( RIGHT$
     ( STR$ (R),1)): VTAB S + 10:
     HTAB 1: PRINT "<"S">";: NEXT
2450  VTAB 23: HTAB 1
2460  IF SE = 4 THEN  PRINT "<S>
     STOP    <#> LOAD      <RTN>
     CONTINUE";
2470  IF SE = 6 THEN  PRINT "<S>
     STOP    <#> DELETE    <RTN>
     CONTINUE";
2480  IF SE = 7 THEN  PRINT "<S>
     STOP                  <RTN>
     CONTINUE";
2490  GOSUB 3190
2500  IF X = 83 THEN  POP : RETURN
2510  IF X = 13 THEN 2410
2520  IF X < 48 OR X > 57 OR SE =
     7 THEN 2490
2530 Q = 10 * ( VAL ( LEFT$ ( STR$
     (R),1))) + X - 48: IF R < 10
      THEN Q = X - 48
```

全能標貼製作器

```
2540  VTAB 10: HTAB 1: CALL  - 95
  8
2550  FOR II = 0 TO 9: HTAB 5: PRINT
  K$(Q,II): NEXT
2560  RETURN
2570  REM ==============
2580  REM READ FROM DISK
2590  REM ==============
2600  IF K$(0,0) <  > "" OR N =  -
  1 THEN  RETURN
2610  ONERR  GOTO 2770
2620  VTAB 10: HTAB 1: CALL  - 95
  8: VTAB 15: HTAB 12: PRINT "
  PLEASE WAIT....."
2630  PRINT D$"OPEN LABEL.DATA"
2640  PRINT D$"READ LABEL.DATA"
2650  INPUT N: IF N =  - 1 THEN 2
  740
2660  FOR J = 0 TO N
2670  FOR I = 0 TO 9
2680 A$ = ""
2690  GET B$: IF  ASC (B$) < 32 THEN
  2690
2700  IF B$ <  >  CHR$ (92) THEN
  A$ = A$ + B$: GOTO 2690
2710 K$(J,I) = A$
2720  NEXT I
2730  NEXT J
2740  PRINT : PRINT D$"CLOSE LABE
  L.DATA"
2750  ONERR  GOTO 150
2760  RETURN
2770  CALL  - 3288: PRINT : PRINT
  D$"CLOSE"
2780  POKE 216,0: VTAB 15: HTAB 1
  2
2790  PRINT  CHR$ (7)" *** ERROR
   *** " CHR$ (7): PRINT : PRINT

2800  PRINT "CHECK YOUR DISK AND
  PRESS <RETURN> TO": PRINT "T
  RY AGAIN."
2810  GET W$: PRINT
2820  VTAB 15: HTAB 1: CALL  - 95
  8: GOTO 2610
2830  REM ========
2840  REM WARNING!
2850  REM ========
2860  VTAB 10: HTAB 1: CALL  - 95
  8: VTAB 13
2870  HTAB 12: INVERSE : PRINT "
  *** WARNING *** ": NORMAL : PRINT
  : PRINT
2880  HTAB 5: PRINT "THIS WILL RE
  PLACE CURRENT LABEL": PRINT

2890  HTAB 13: PRINT "OK TO PROCE
  ED ? ";: GOSUB 3210
2900  IF X = 78 THEN  POP : RETURN

2910  FOR II = 1 TO 10:L$(II) = "
  ": NEXT
2920  RETURN
2930  REM ===============
2940  REM FIRST TIME USE?
2950  REM ===============
2960  ONERR  GOTO 3020
2970  PRINT : PRINT D$"OPEN LABEL
  .DATA"
2980  PRINT D$"READ LABEL.DATA"
2990  INPUT N
3000  PRINT D$"CLOSE LABEL.DATA"
3010  POKE 216,0: RETURN
3020  PRINT D$"CLOSE": IF  PEEK (
  222) <  > 5 THEN 3070
3030  PRINT D$"OPEN LABEL.DATA": PRINT
  D$"WRITE LABEL.DATA"
3040 N =  - 1: PRINT N
3050  PRINT D$"CLOSE LABEL.DATA"
3060  POKE 216,0: PRINT  CHR$ (7)
  : PRINT  CHR$ (7): GOTO 130
3070  PRINT "ERROR "; PEEK (222):
  PRINT "OCCURRED WHILE TRYIN
  G TO INITIALIZE": PRINT "YOU
  R DATA FILE": PRINT : PRINT
  "PLEASE CORRECT THE PROBLEM
  AND PRESS": PRINT "ANY KEY T
  O START AGAIN": GET K$: RUN

3080  REM ===============
3090  REM CHARACTER INPUT
3100  REM ===============
3110 A$ = ""
3120  GET B$
3130  IF B$ =  CHR$ (13) THEN  CALL
  - 868: RETURN
3140  IF B$ <  >  CHR$ (8) AND  LEN
  (A$) = 32 THEN  PRINT  CHR$
  (7);: GOTO 3120
3150  IF B$ <  >  CHR$ (8) THEN A
  $ = A$ + B$: PRINT B$;: GOTO
  3120
3160  IF  LEN (A$) = 0 THEN 3120
3170  PRINT B$"."B$;: IF  LEN (A$
  ) = 1 THEN A$ = "": GOTO 312
  0
3180 A$ =  LEFT$ (A$, LEN (A$) -
  1): GOTO 3120
3190 X =  PEEK ( - 16384): IF X <
  128 THEN 3190
3200  POKE  - 16368,0:X = X - 128
  : RETURN
3210 X =  PEEK ( - 16384): IF X <
  128 OR (X <  > 217 AND X <  >
  206) THEN 3210
3220  POKE  - 16368,0:X = X - 128
  : RETURN
3230  FOR I = 1 TO 10: HTAB 5: PRINT
  L$(I): NEXT : RETURN
3240  PRINT "
                                        ";: RETURN
  : REM 40 SPACES BETWEEN QUOT
  ES
```

```
10  REM  *************************
20  REM  *   SET UP PRINTER     *
70  REM  *************************
80  TEXT : POKE 34,0: POKE 216,0:
     HOME
90 D$ =  CHR$ (4)
100  HTAB 15: INVERSE : PRINT " S
     ET UP PRINTER": NORMAL : POKE
     34,2: VTAB 4
110  PRINT "THIS PROGRAM WILL MOD
     IFY 'POSTMASTER'": PRINT "SO
      THAT THE PROGRAM WILL WORK
     ON YOUR": PRINT "PRINTER."
120  PRINT : PRINT "YOUR PRINTER
     MUST BE CAPABLE OF": PRINT "
     PRINTING ENHANCED, CONDENSED
     , AND": PRINT "ENLARGED TYPE
     ."
130  PRINT : PRINT "TO PROCEED, Y
     OU WILL NEED A COPY OF": PRINT
     "'SET UP PRINTER' AND 'LABEL
      MACHINE' ON A NON-": PRINT
     "WRITE PROOTECTED DISK, AND
     YOUR": PRINT "PRINTER MANUAL
     ."
140  VTAB 21: PRINT "**  COPYRIGH
     T 1984 BY COMPUTING AGE   **"
     : VTAB 23: PRINT "    HIT AN
     Y KEY WHEN READY ..... ";: GET
     W$: HOME
150  PRINT "    (EXAMPLE AT BOT
     TOM IS FOR": PRINT "        S
     TANDARD PARALLEL CARD,": PRINT
     "       EPSON MX-80 PRINTER"
     : POKE 34,10
160  HOME : PRINT "ENTER THE CODE
      FOR MAXIMUM": PRINT "CHARAC
     TERS PER LINE:": VTAB 23: PRINT
     "CHR$(9)+" CHR$ (34)80"N" CHR$
     (34): VTAB 14: INPUT "";I$
170  HOME : PRINT "ENTER THE CODE
      FOR TURNING": PRINT "ENHANC
     EMENT ON:": VTAB 23: PRINT "
     CHR$(27)+CHR$(69)": VTAB 14:
      INPUT "";T1$
180  PRINT : PRINT : PRINT "ENTER
      THE CODE FOR TURNING": PRINT
     "ENHANCEMENT OFF:": VTAB 23:
      PRINT "CHR$(27)+CHR$(70)": VTAB
      20: INPUT "";C1$
190  HOME : PRINT "ENTER THE CODE
      FOR TURNING": PRINT "ENLARG
     EMENT ON:": VTAB 23: PRINT "
     CHR$(14)": VTAB 14: INPUT ""
     ;T2$
200  PRINT : PRINT : PRINT "ENTER
      THE CODE FOR TURNING": PRINT
     "ENLARGEMENT OFF:": VTAB 23:
      PRINT "CHR$(20)": VTAB 20: INPUT
     "";C2$
210  HOME : PRINT "ENTER THE CODE
      FOR TURNING": PRINT "CONDEN
     SED CHARACTERS ON:": VTAB 23
     : PRINT "CHR$(15)": VTAB 14:
      INPUT "";T3$
220  PRINT : PRINT : PRINT "ENTER
      THE CODE FOR TURNING": PRINT
     "CONDENSED CHARACTERS OFF:":
      VTAB 23: PRINT "CHR$(18)": VTAB
      20: INPUT "";C3$
230  POKE 34,2: HOME
240  VTAB 5: PRINT "        PLEASE
      DOUBLE CHECK INPUT": PRINT
     : PRINT : PRINT
250  PRINT "MAX CHARS:";: HTAB 20
     : PRINT I$
260  PRINT "ENHANCEMENT ON:";: HTAB
     20: PRINT T1$
270  PRINT "              OFF:";: HTAB
     20: PRINT C1$: PRINT
280  PRINT "ENLARGEMENT ON:";: HTAB
     20: PRINT T2$
290  PRINT "              OFF:";: HTAB
     20: PRINT C2$: PRINT
300  PRINT "CONDENSED ON:";: HTAB
     20: PRINT T3$
310  PRINT "            OFF:";: HTAB
     20: PRINT C3$
320  VTAB 23: HTAB 1: PRINT "ARE
     YOU SURE THIS IS CORRECT? (Y
     /N) ";: GET W$: IF W$ <  > "
     Y" AND W$ <  > "N" THEN 320
330  IF W$ = "N" THEN  RUN
340  HOME : VTAB 8: PRINT "MAKE S
     URE THAT 'POSTMASTER' IS ON
     THE": PRINT : PRINT "DISK IN
      THE CURRENT DRIVE.": PRINT
     : PRINT : PRINT
350  PRINT "HIT ANY KEY TO START.
     .... ";: GET W$: PRINT : POKE
     34,20: VTAB 23
360  FOR I = 1 TO 40
370  IF  MID$ (I2$,I,3) =  CHR$ (
     34) + "X" +  CHR$ (34) THEN
     I2$ =  LEFT$ (I2$,I - 1) + "
     CC$" +  RIGHT$ (I2$, LEN (I2
     $) - I - 2): GOTO 390
380  NEXT I
390  PRINT D$"OPEN INSTALL.EXEC"
400  PRINT D$"WRITE INSTALL.EXEC"
410  PRINT "LOAD PERFECT LABEL CR
     EATING MACHINE"
420  PRINT "1070 I$=";I$
430  PRINT "1080 T1$=";T1$
440  PRINT "1090 C1$=";C1$
450  PRINT "1100 T2$=";T2$
460  PRINT "1110 C2$=";C2$
470  PRINT "1120 T3$=";T3$
480  PRINT "1130 C3$=";C3$
490  PRINT "SAVE POSTMASTER"
500  PRINT "POKE 34,0"
510  PRINT "HOME"
520  PRINT "DELETE INSTALL.EXEC"
530  PRINT D$"CLOSE INSTALL.EXEC"
540  PRINT D$"EXEC INSTALL.EXEC"
```

我們經常會面對許多考驗，需要花精神去思索決定。本文提供一個相當獨特的軟件給予成功人士及管理行政階層人士作爲一種輔助決策的工具。當然，任何人都可應用到這個系統的優點。它的用途是隨你的想像力伸展。

寧峯

行政思考決策輔助系統

成功人士必備的工具 幫助作出明智的抉擇

各位讀者，今次爲大家大胆嘗試介紹一個非常實用的程式——EXECUTIVE ANALIZER 。

EXECUTIVE ANALIZER 是一個幫助商業及私人上問題的「解決工具」。它能通過分析能力，而作出決策。此程式會將影响決策的因素分開幾個部份，而當設定好各種因素後，程式會要求告知每一個因素之重要性。接着，程式會從多方面衡量而做出最後決定。

程式分析結果是以分數顯示，電腦會顯示出每一因素每一方面的分析結果。當然，用家可將結果用印字機印出或儲存入磁碟上。

EXECUTIVE ANALIZER 是適用於多種商業及私人决策上的問題，例如：

1）投資分析及比較
2）房屋投資
3）買車
4）個人興趣的比較
5）估計未來的僱主們
6）計劃分析職業之價値
7）資金設備之獲得
8）地產之獲得
9）估計分析僱員
10）估計及分析工作表現
11）其他……

而此程式之設計，有幾個主要的特徵：

1）用家在目錄的選擇上，無須鍵入任何數字。（操縱方法後述）

2）在同一時間，程式最多可以從五個不同角度去分析二十個因素。

3）每一個決擇（ALTERNATIVE）及因素之名稱可以長達二十五個字。

4）决擇及各因素，均可隨意更改或刪除。

5）如前所述，每一個因素，都能用－100 至100 間的數顯示出其相對重要性。

6）另外，每一個决擇也可以由數字顯示出其等級。

7）最後一點，分析結果會被繼續修訂，用以顯示改變因素或優先次序所帶來之影响。

中斷系統

注意：當程式運行中時，千萬不要按“RESET”KEY或CTRL－C 鍵。如你按下，電腦則會詢問是否眞的要中斷程式工作。這時你可能可以恢復工作，但結果如何則是不可預測的，但如果你決定中斷程式或在程式運行中電腦發現有錯誤出現時，則程式會自動將原先輸入的資料以ADUMP爲名儲存在磁碟上。

程式概念：

在使用此程式前，用家必須首先明白程式的五個基本概念，分別是決擇之選擇（ALTERNATIVES）、因素（FAC－TORS）、等級（GRADES）及分析衡量因素的技巧（WEIGHTED FACTOR ANALYSIS TECHNIQUE）。

㈠抉擇因素

在一些複雜的問題，我們時常要作出多方面的分析選擇。例如：當你要諮詢一些雜誌廣告或參考一些新產品的圖表時，你是需要作出一些選擇的，而當你的決擇因素爲五個或以下時，你便可使用此程式。而當你的決擇因素設定後，你便可以進行下一步工作——將你的選擇冠上名字。（但不能超過二十五字）

㈡因素之設定及選擇與優先權的問題

在你對某件事作出決定時，你必須收集及預備好一張對事件有影响的因素列表。例如，當你要買屋時，影响你决定的因素可以有價錢、地點、大小及其他。

這裏提供幾點選擇因素的要點，以作參考：

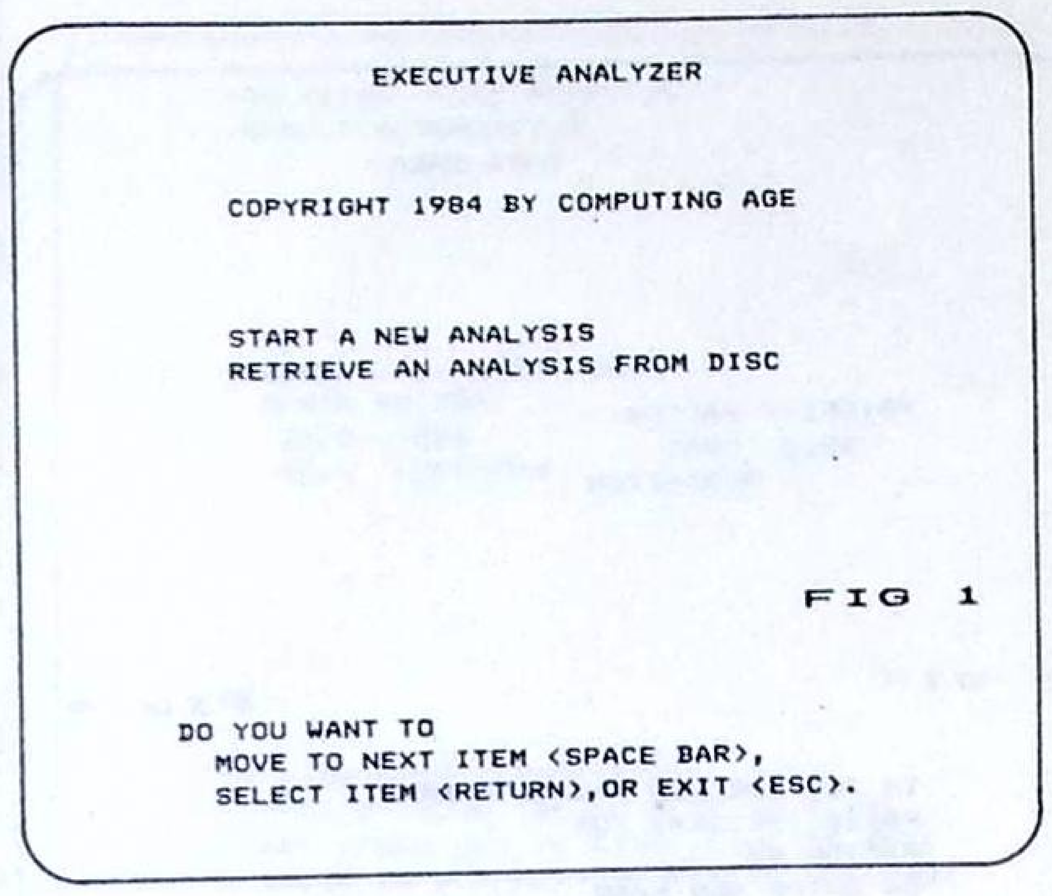
1）刪去那些並不影响決擇的因素。
2）刪去那些初時想到的，但現在並不覺得重要因素。
3）如果發現有些因素是有關連的，你可以將它們集合一起，成爲一個因素。
4）找出那些表面上看來不同，但實質上是一樣或類似的因素，將它們適當地刪去或集合一起。

而此程式是容許你有二十個或以下的因素存在的。

現在我們開始討論優先權的問題。由於並不是每一個因素都是同等重要的，故優先權的問題應運而生。首先，我們要決定以什麼爲標準來排列因素的先後。

這裏提供了一些資料以供參考：

1）界定重要及次要因素。例如，用“IMPORTANT”及



表(一)

FACTOR NAME	PRIORITY(P)	ALTERNATIVE1 GRADE(G)	P*G	ALTERNATIVE2 GRADE(G)	P*G
FACTOR A	1.0	75.0	75.0(1)	40.0	40.0(1)
FACTOR B	5.0	30.0	150.0(1)	40.0	200.0(1)
FACTOR C	10.0	50.0	500.0(1)	60.0	600.0(1)
SUM OF P	16.0(3)	SUM OF P*G	725.0(2)		840.0(2)
SCORE(SUM OF P*G/SUM OF P)			45.3(4)		52.5(4)

"NOT IMPORTANT" ，或用"VERY IMPORTANT" ，"IMPORTANT"及"NOT SO IMPORTANT"三個等級。

2）由於這是一個「數字的比例形式」（NUMERIC RAT—ING SCALE ，我們可以用大數或小數代表最重要的因素。

3）事前必須選定一個數字符號。例如，如只是有二個等級，可用0，1或1，2。

而此程式是容許用家在—100至100之間選擇數字範圍的。

㈢等級

當你完成上述的工作後，你便要將決擇因素分等級。

此過程分三個步驟：①選定等級比例、②選定等級判斷的標準、③衡量、分析一個決擇因素。

①選定等級比例：

首先你要選定最好及最壞的決擇之界限。例如："ADE—QUATE"及"NOT ADEQUATE" ，或者"VERY ADEQUATE"，"ADEQUATE"及"NOT SO ADE—QUATE"。另一方面，你也可以用"VERY GOOD"到"VERY POOR"爲範圍。

在這裏你要選定以高分數或低分數代表最好的。但注意，如果你選擇高分數表最好的的話，則有關優先權之界定必須一樣，相反亦然。

同樣的，你可以於—100至+100 間選擇數字的範圍。

② 選定等級判斷的標準：

這時，你必須決定一個決擇對於一個因素之好壞等級。是好？是中等？或次等的。例如：買屋，比例是0～10，0代表最差，10代表最好。在一個名爲"NO. OF BEDROOM"的因素下，你可能決定如一屋只有一或兩間睡房，則給予第2等級，如有三間則以第6等級表之，如此類推。

③ 分析一個決擇因素：

當你完成了上述兩個步驟後，接下來，便要比較那些決擇及給予一個等級於每一因素。你有兩種途徑可用之。第一：你可以將每一個因素之決擇因素相對地比較，並給予等級。而另一方法，則是抽出一些獨立的決擇，再分別繼續分析。

㈣分析及衡量因素

此乃程式最重要之一環，此程式會用用家輸入的資料去分析及衡量一個或多個因素。此種技巧主要來說有四個步驟、要點。注意：下列所指的某某倍數、相加是指其用以分析之數字而言。

1）所有因素之決擇因素的等級均是因素優先權之倍數。

2）所有因素之優先權是加在一起的。

3）優先權之總和是分爲"GRADES PRIORITY PRODU—CT"之總和。

4）而所有因素之"GRADES PRIORITY PRODUCT 也是加在一起的。

表一將每一步驟清楚地指明。

在此例子，優先權比例之範圍是由（1.0～10.0）、1.0是表"次要的"，而10.0 是表"重要的"。而等級比例之範圍則由（0.0～100.0），0.0表"最差的"，而100.0表"最好的"。至於分析結果之分數顯示範圍與等級比例範圍一樣。（參考表一）

當分析得出結論時，觀察其結果是否和你期望的一樣。而由於此程式是根據一個途徑去分析的，故你是可以回去看你輸入的資料。而且你更加將資料加以修改，從而看出及比較其結論。現提供幾點有關估計衡量一結果應注意之事項：

1）找出那些因素影响結論最大，因這些因素之優先權等級應是特高或者偏向"勝利者"。（指結論）

2）找出那些因素最不影响結論。而這些因素之優先權等級應是奇低或者偏向"失敗者"的。

3）在任何情況下，都必須注意到"個人偏好"（PERSON—AL BIASES）。因爲所有資料都是根據個人所好輸入的。如果有幾個人一起衡量的話，留意同一因素、分析角度之不同。因此能代表個人偏好或更加瞭解其他參與者。

系統概念

當你鍵入程式後，如鍵入無誤則會出現如圖一的畫面，此乃程式之主要目錄。

而當你輸入所有需要分析資料並得出結論後，此目錄會被擴展，容許你改變分析或用印字機印出。（參考圖二）

如前所述，此程式有一特點——在目錄上作選擇時不需輸入數字。目錄上左邊會有一個游標出現，用SPACE BAR即令其上下移動，如想執行，只需按RETURN 。而如想改變已選定之資料時，可按ESC重新選擇。

```
ANALYSIS:  HOME PURCHASE
 RESULT ALTERNATIVE
         ALTERNATIVE #1
         ALTERNATIVE #2

   START A NEW ANALYSIS
   RETRIEVE AN ANALYSIS FROM DISC
   ADD AN ALTERNATIVE
   DELETE AN ALTERNATIVE
   CHANGE AN ALTERNATIVE'S NAME
   ADD A FACTOR
   DELETE A FACTOR
   CHANGE A FACTOR'S PRIORITY
   CHANGE A FACTOR'S NAME
   CHANGE AN ALTERNATIVE'S GRADE
   PRINT ANALYSIS

DO YOU WANT TO
  MOVE TO NEXT ITEM <SPACE BAR>,
  SELECT ITEM <RETURN>,OR EXIT <ESC>.
```

FIG 2

```
ANALYSIS:  HOME PURCHASE
  VALUE PARAMETERS
    3.0  NUMBER OF ALTERNATIVES
    3.0  NUMBER OF FACTORS
  100.0  HIGHEST FACTOR PRIORITY
    1.0  LOWEST FACTOR PRIORIT
  100.0  HIGHEST POSSIBLE SCORE
    1.0  LOWEST POSSIBLE SCORE
    0.0  SET ALTERNATIVES FIRST(0)
         OR  FACTORS FIRST (1)

TO CHANGE A PARAMETER,
  CHANGE VALUE (ARROW KEYS TO START,
  ANY OTHER KEY TO STOP),MOVE TO NEXT
  VALUE <SPACE KEY>,OR FINISH <RETURN>
```

FIG 3

```
PRIORITY FACTOR
   50.0  COST
         ?LOCATION

TO SET THE NEW FACTOR'S NAME,
PRESS <RETURN> FOR
FACTOR #2
OR ENTER NEW NAME.
```

FIG 4

另一方面，你也不需要輸入所需之數字，只需用“←→”即可使數字自動增加及減少。（←：減少；→：增加），而如要停止增加或減少，只需按任何一鍵卽可。

而當你完成工作後，按ESC 卽可離開程式，但程式會先問你是否需要存起剛剛之工作資料。

但電腦是會顯示出一個預設名字，而程式更提供“C”鍵爲CATALOG 功能，使你觀看CATALOG。注意，如輸入自己的名稱是不可超過25字的。接着，程式會再問你是否要EXIT ，如你決定EXIT ，則程式會再顯示出主目錄。

系統的操作

㈠開始工作

㈠開始工作：在主要目錄上，如你選擇“START A NEW ANALYSIS”，你會被要求輸入名稱。

㈡設定主要參數

㈡設定主要參數：接着，螢幕會被轉換並要求用家設定主要參數。在此我們會用一非常簡單的例子作爲樣本供各位參考。（參考圖三）

但注意，優先權及等級比例一經選定，便不能增加了。而另一方面因素及决擇因素之數目是可以更改的。但記着，程式之決擇數目是由２～５個，因素則是１～20個，而優先權及等級比例之數目是由一100～100的。

〔開始時，你是會被要求選擇建立决擇之資料先或因素資料。其主要分別是如你選擇先建立分析角度資料的話，則你會被要求一次輸入一個因素之所有決擇因素，而如你選擇先設定因素資料的話，則會被要求一次輸入同一個決擇因素於所有因素。

現在，我們會分別討論２種選擇之工作過程。〕

㈢先設定因素資料

當你輸入所有之參數後，按〔RETURN〕會要求輸入第一個因素名稱，而程式是會提供預設名字的，如你不喜歡此名稱的話，也可自己輸入。（參考圖四）之後是設定優先權之值，此時程式會提供預設值給你，當然你亦可以自己輸入。接着，你要輸入第一個决擇因素之名稱。接着你要根據相對之因素給予等級。跟着再重覆上述步驟直至達到所選之目標。（參考圖五）

當上述步驟工作後，主要目錄表會再出現，這時你可增加資料或再開始另一個。

㈣選定抉擇因素

如你決定先選定决擇因素時，則程式會要求你輸入所有的决擇之名稱，方法請參照上段。接着再輸入第一個因素之名稱及優先權。（參考圖六）

這時，你會被要求根據剛輸入之因素比較所有的决擇因素及給予一個等級。完成後按〔RETURN〕，則會重複上述步驟。（參考圖七）

㈤從磁碟中取回分析

要達到此目的，首先要將主要目錄上之游標移動到“RETRIEVE AN ANALYSIS FROM DISK”，按“RETURN”。此時，如果你曾經修改過一個分析資料或曾開始一個分析資料，則程式會問你是否需要先存起剛剛使用之資料。

這樣你剛輸入之分析資料則會被存入磁碟上。

接着，你可以輸入欲取出之資料檔案名稱，當LOAD完後，這個檔案之名稱、决擇因素之名稱及分數都會在螢幕頂端顯示出來。

㈥增加抉擇因素

如你選擇“ADD AN ALTERNATIVE”，則程式會首先檢查你的决擇因素是否超過限額（５個），如沒有則容許你輸入名稱。（參考圖八） 之後，則重複上述設定决擇因素之工作。（參考圖九）

㈦刪去抉擇因素

如你選擇“DELETE AN ALTERNATIVE”，則程式首先會檢查你已有的决擇因素有幾多個，是否有兩個或以上。

此時如你按〔RETURN〕，則會刪去ALTERNATIVE ＃１，但如你改變主意的話，可按〔ESC〕，則程式並不會修改任何資料。

㈧改變抉擇之名稱

如你選擇“CHANGE AN ALTERNATIVE'S NAME”，則第一步便是選擇要更改名稱之决擇因素，選定後，輸入新名稱卽可。但如你改變主意的話，而又未輸入任何新名稱

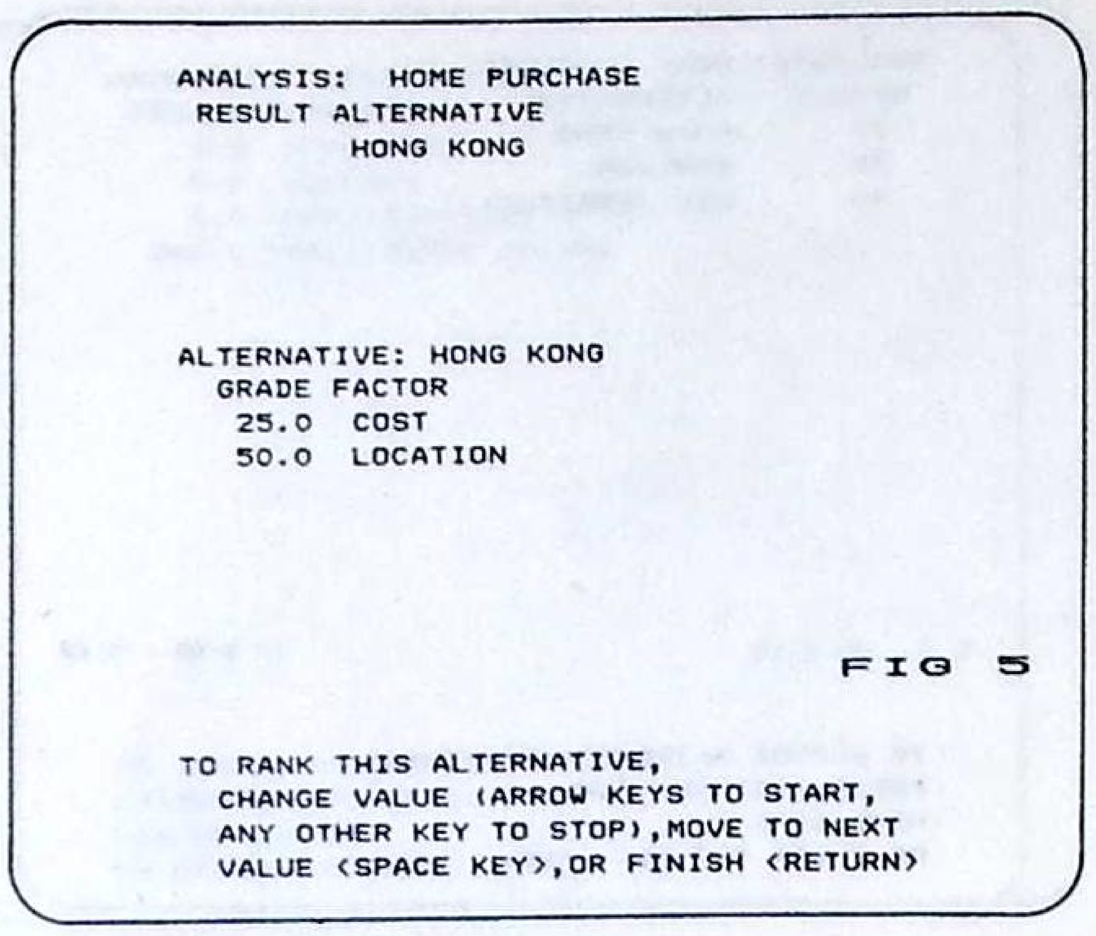


FIG 5

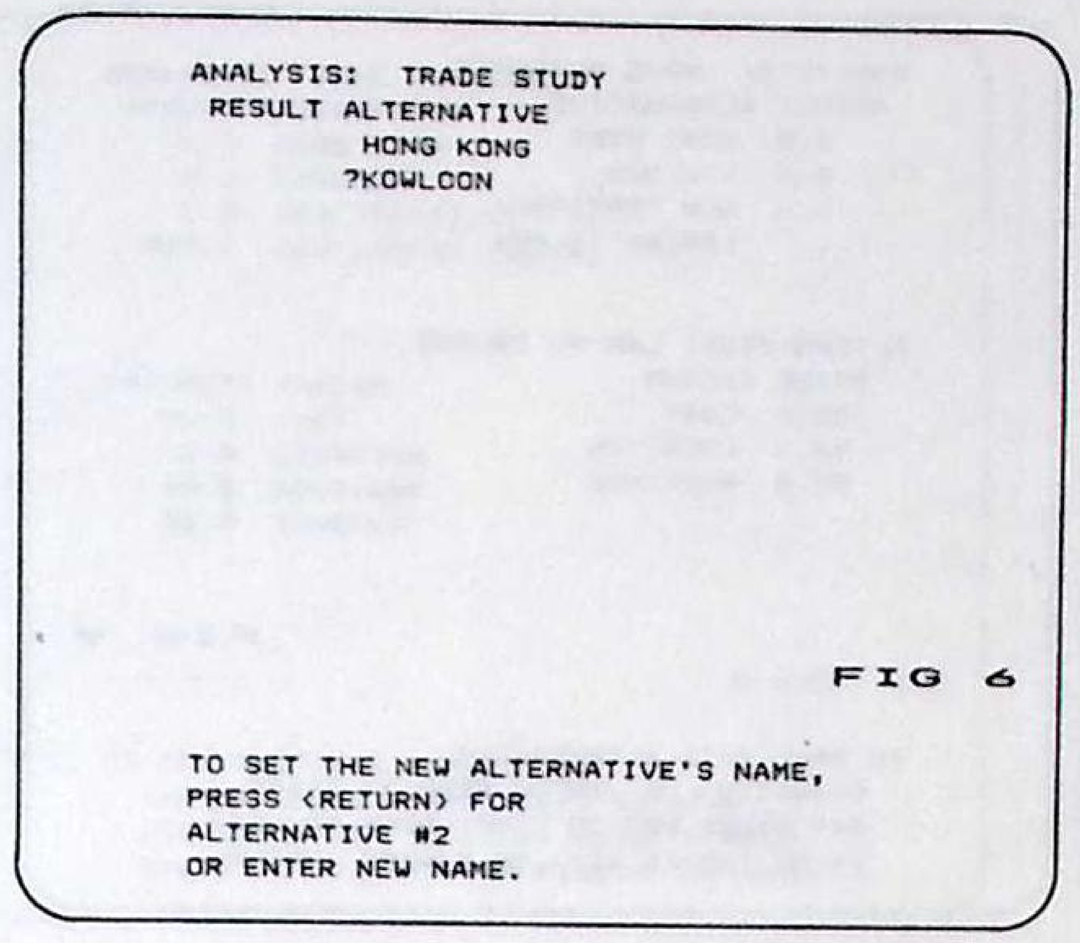


FIG 6

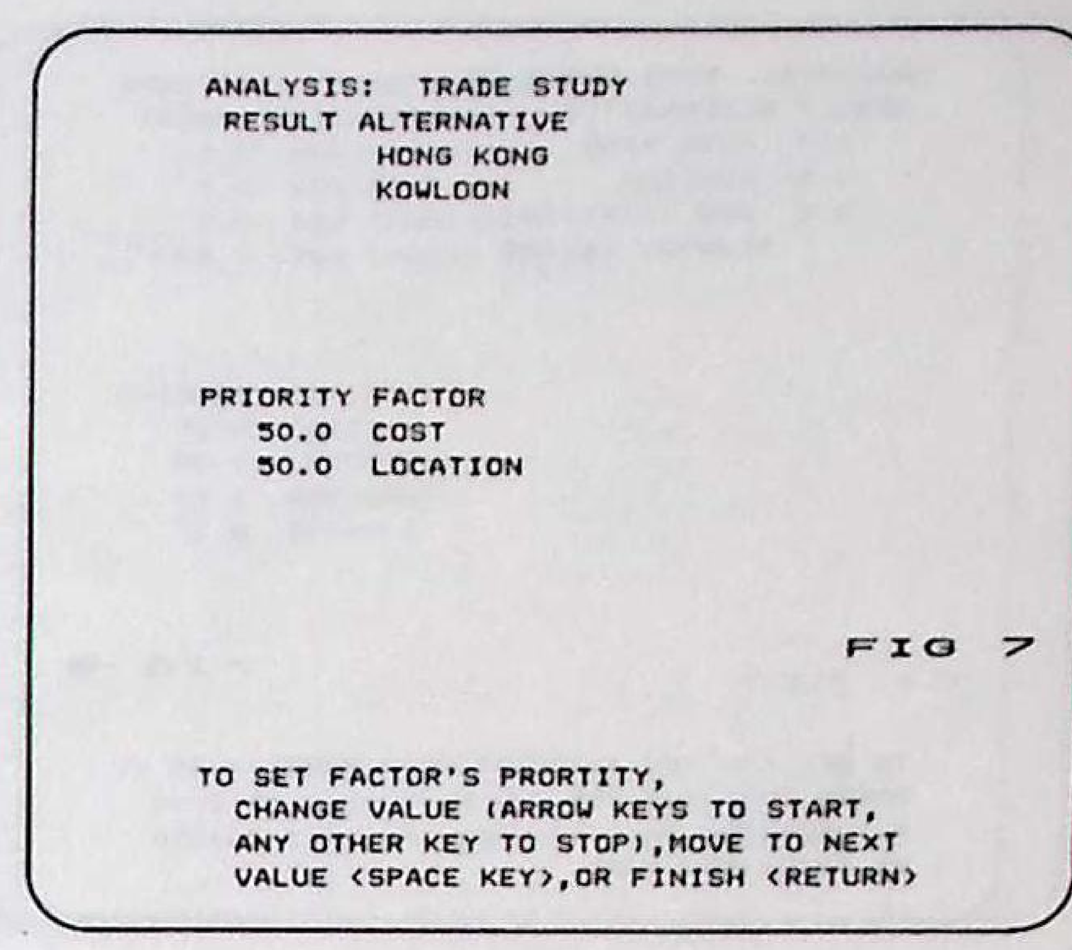


FIG 7

時，按〔RETURN〕則會保存原有的名字。（參考圖十及圖十一）

增加因素

如你選擇〝ADD A FACTOR〞，則程式會首先檢查你已有的因素是否已超過二十個，如沒有，則容許你輸入新因素之名稱，之後程式會回到第三步及第四步（設定優先權等之資料）。（參考圖十二）

(十)刪去因素資料

如你選擇〝DELETE A FACTOR〞，則程式會首先檢查你是否有2個或以上之因素，如有則容許你刪去一個因素。（參考圖十三）

選定後按〔RETURN〕則刪去此因素。但如你改變主意，則按ESC 可令程式回到主要目錄表，而不作任何修改。

(土)改變因素的優先權

只要你選擇〝CHANGE A FACTOR'S PRIORITY〞，則容許你更改一個因素之優先權。而更改後，一切之分析結果都會被更新及顯示。（參考圖十四）

(圭)改變因素之名稱

如你選擇〝CHANGE A FACTOR'S NAME〞，則你首先需要選定一個要更改之因素，然後輸入新名稱，但如你改變主意，而又未輸入任何新名稱的話，則按〔RETURN〕會保存原有之名稱而回到主要目錄上。（參考圖十五）

(亖)改變抉擇因素之等級

當你選擇〝CHANGE AN ALTERNATIVE'S GRADE〞，你首先要選定一個決擇因素，選定後，則會顯示出其本來之等級，更改後，當然，所有的分析結果也會被更新及顯示。（參考圖十六）

(亗)印出結果

選定〝PRINT ANALYSIS〞，則容許你印出分析於印字機上。首先，你要輸入印字機介面咭之SLOT 位置（1～5。

```
]LIST

10  REM ************************
    *
20  REM *                      
    *
30  REM *  EXECUTIVE ANALYZER  
    *
40  REM *                      
    *
50  REM *  COPYRIGHT (C) 1984  
    *
60  REM *  BY COMPUTING AGE    
    *
70  REM *                      
    *
80  REM *                      
    *
90  REM ************************
    *
100 GOTO 6700
110 REM * UTILITIY ROUTINES****
120 REM * URDATE ALT.RESULTS*
130 GOSUB 5440: PRINT "*** UPDAT
    ING SCORES ***"
140 IF PSUM% <  > 0 THEN 160
150 VLOC% = 2: FOR J = 1 TO ANUM%
    :ARES(J) = 0.0:VLOC% = VLOC%
     + 1:NUMB = ARES(J): GOSUB 1
    80: NEXT : RETURN
160 FOR J = 1 TO ANUM%:GSUM%(J) =
    0: NEXT : FOR J = 1 TO FACT%
    : FOR K = 1 TO ANUM%:GSUM%(K
    ) = GSUM%(K) + PDES%(J) * GD
    ES%(K,J): NEXT : NEXT :VLOC%
     = 2: FOR J = 1 TO ANUM%:ARE
    S(J) = GSUM%(J):VLOC% = VLOC
    % + 1:NUMB = ARES(J): GOSUB
    180: NEXT : RETURN
170 REM * PRINT A REAL*******
180 A =  INT (NUMB) +  INT ((NUMB
     -  INT (NUMB)) * 10 + 0.5)
    10:A% = 8: IF A =  INT (NUMB
    ) OR A - 1 =  INT (NUMB) THEN
    A% = 6
190 IF PAGE% = 0 THEN  VTAB VLOC
    %: HTAB 2
```

```
ANALYSIS:  HOME PURCHASE
 RESULT ALTERNATIVE
    0.0  HONG KONG
    0.0  KOWLOON
    0.0  NEW TERRITORY
        ?LANTAU ISLAND

                                FIG 8

TO SET THE NEW ALTERNATIVE'S NAME,
PRESS <RETURN> FOR
ALTERNATIVE #4
OR ENTER NEW NAME.
```

```
ANALYSIS:  HOME PURCHASE
 RESULT ALTERNATIVE
    0.0  HONG KONG
    0.0  KOWLOON
    0.0  NEW TERRITORY
         LANTAU ISLAND

ALTERNATIVE: LANTAU ISLAND
  GRADE FACTOR
   50.0  COST
   50.0  LOCATION
   50.0  MORTGAGE

                                FIG 9

TO RANK THIS ALTERNATIVE,
  CHANGE VALUE (ARROW KEYS TO START,
  ANY OTHER KEY TO STOP),MOVE TO NEXT
  VALUE <SPACE KEY>,OR FINISH <RETURN>
```

```
ANALYSIS: HOME PURCHASE
 RESULT   ALTERNATIVE
  20      HONG KONG
  30      KOWLOON
  40      NEW TERRITORY

                                FIG 10

TO CHANGE ALTERNATIVE'S NAME,
PRESS <RETURN> FOR
HONG KONG
OR ENTER THE NEW NAME.
```

```
200  PRINT  TAB( A% -  LEN ( STR$
     (A)));A;
210  IF A% = 6 THEN  PRINT ".0";
220  RETURN
230  REM  * PRINT A NAME ***
240  HTAB 10: PRINT INPT$;
250  IF PAGE% <  > 0 THEN  PRINT

260  RETURN
270  REM *CLEAR WINDOW***
280  IF WNDW% < 0 OR WNDW% > 4 THEN
      RETURN
290  IF WNDW% = 0 THEN 390
300  ON WNDW% GOTO 310,330,350,37
     0
310 VFIR% = 1:VBOT% = 24
320  GOTO 390
330 VFIR% = 1:VBOT% = 8
340  GOTO 390
350 VFIR% = 9:VBOT% = 20
360  GOTO 390
370 VFIR% = 21:VBOT% = 24
380  GOTO 390
390  VTAB VFIR%
400  POKE 34,VFIR% - 1
410  POKE 35,VBOT%
420  HOME
430  RETURN
440  REM  * DISP ALT. & RESULTS*
450  IF PAGE% <  > 0 THEN 480
460 WNDW% = 2: GOSUB 280
470 VLOC% = VFIR% + 1
480  PRINT "ANALYSIS:  ";NAME$
490  PRINT " RESULT ALTERNATIVE"
500  IF ANUM% < 1 THEN  RETURN
510  FOR J = 1 TO ANUM%
520 NUMB = ARES(J)
530 VLOC% = VLOC% + 1
540  GOSUB 180
550 INPT$ = ADES$(J): GOSUB 240
560  NEXT
570  RETURN
580  REM  * DISP FACTORS & PRIORI
     TES
590  IF PAGE% <  > 0 THEN 620
600 WNDW% = 3: GOSUB 280
610 VLOC% = VFIR% + 1
620  PRINT
```

```
630  PRINT "PRIORITY FACTOR"
640  IF FACT% < 1 THEN  RETURN
650 A% = FCUR% + FLNG% - 1: IF A%
     > FACT% THEN A% = FACT%
660  FOR J = FCUR% TO A%
670 NUMB = PDES%(J)
680 VLOC% = VLOC% + 1
690  GOSUB 180
700 INPT$ = FDES$(J): GOSUB 240
710  NEXT
720  RETURN
730  REM *DISP FACS & GRADES*
740 WNDW% = 3: GOSUB 280
750 VLOC% = VFIR% + 1
760  PRINT "ALTERNATIVE: ";ADES$(
     ACUR%)
770  PRINT "  GRADE FACTOR"
780  IF FACT% < 1 THEN  RETURN
790 A% = FCUR% + FLNG% - 1: IF A%
     > FACT% THEN A% = FACT%
800  FOR J = FCUR% TO A%
810 NUMB = GDES%(ACUR%,J)
820 VLOC% = VLOC% + 1
830  GOSUB 180
840 INPT$ = FDES$(J): GOSUB 240
850  NEXT
860  RETURN
870  REM  *DISP ALT.&GRADES *
880  IF PAGE% <  > 0 THEN 910
890 WNDW% = 3: GOSUB 280
900 VLOC% = VFIR% + 1
910  PRINT "FACTOR: ";FDES$(FCUR%
     )
920  PRINT "  GRADE  ALETERNATIVE
     "
930 A% = ACUR% + ALNG% - 1: IF A%
     > ANUM% THEN A% = ANUM%
940  FOR J = ACUR% TO A%
950 NUMB = GDES%(J,FCUR%)
960 VLOC% = VLOC% + 1
970  GOSUB 180
980 INPT$ = ADES$(J): GOSUB 240
990  NEXT
1000  RETURN
1010  REM  * GET A NAME ***
1020  POKE 51,191
1030 INPT$ = ""
1040  CALL  - 662
```

```
ANALYSIS:  HOME PURCHASE
 RESULT ALTERNATIVE
    0.0  HONG KONG
    0.0  KOWLOON
    0.0  NEW TERRITORY
 000.0 ?NEW LANTAU ISLAND

TO CHANGE ALTERNATIVE'S NAME,
PRESS <RETURN> FOR
LANTAU ISLAND
OR ENTER NEW NAME.
```

FIG 11

```
ANALYSIS:  HOME PURCHASE
 RESULT ALTERNATIVE
    0.0  HONG KONG
    0.0  KOWLOON
    0.0  NEW TERRITORY
  000.0  NEW LANTAU ISLAND

PRIORITY FACTOR
   70.0  COST
   50.0  LOCATION
   40.0  MORTGAGE
   35.0  SCHOOLS

TO SET FACTOR'S PRORTITY,
  CHANGE VALUE (ARROW KEYS TO START,
  ANY OTHER KEY TO STOP),MOVE TO NEXT
  VALUE <SPACE KEY>,OR FINISH <RETURN>
```

FIG 12

```
ANALYSIS:  HOME PURCHASE
 RESULT ALTERNATIVE
    0.0  HONG KONG
    0.0  KOWLOON
    0.0  NEW TERRITORY
  000.0  NEW LANTAU ISLAND

PRIORITY FACTOR
   70.0  COST
   50.0  LOCATION
   40.0  MORTGAGE
   35.0  SCHOOLS

TO DELETE A FACTOR,
  MOVE TO NEXT ITEM <SPACE BAR>,
  SELECT ITEM <RETURN>,OR EXIT <ESC>.
```

FIG 13

```
1050  FOR J = 512 TO 767
1060 A% =  PEEK (J) - 128
1070  IF A% = 13 THEN 1100
1080 INPT$ = INPT$ +  CHR$ (A%)
1090  NEXT
1100  IF  LEN (INPT$) = 0 THEN IN
     PT$ = ALPH$
1110 ALPH$ =  LEFT$ (INPT$,WLNG%)

1120  RETURN
1130  REM  * SELECT FILE NAME ***

1140 WNDW% = 4: GOSUB 280
1150  PRINT A$
1160  PRINT "ENTER ANALYSIS NAME
     ('C' FOR CATALOG;": PRINT "'
     M' FOR MENU;"
1170  PRINT " <RETURN> FOR ";NAME
     $;")";" ";
1180 ALPH$ = NAME$
1190  GOSUB 1020
1200  IF ALPH$ <  > "C" THEN 1250

1220  PRINT CD$;"CATALOG"
1230  GOSUB 5510
1240  GOTO 1140
1250 NAME$ = ALPH$: IF ALPH$ = "M
     " THEN  CALL  - 10621: HOME
     : GOTO 6820
1260 WNDW% = 1: GOSUB 280
1270  RETURN
1280  REM   * WRITE DATA BASE **

1290  IF SFLG% <  > 1 THEN  RETURN

1300 A$ = "SAVE ANALYSIS TO DISC
     -"
1310  GOSUB 5690
1320  IF INPT% <  > 1 THEN  RETURN

1330 A$ = "TO SAVE ANALYSIS,"
1340  GOSUB 1140
1350  GOSUB 5440
1360  PRINT "*** SAVING ";NAME$;"
     ***"
1370  PRINT CD$;"OPEN ";NAME$
1380  PRINT CD$;"WRITE ";NAME$
1390  PRINT ANUM%;",";FACT%;",";P
     HI%;",";PLOW%;",";GHI%;",";G
     LOW%
1400  FOR J = 1 TO ANUM%
1410  PRINT ADES$(J)
1420  PRINT ARES(J)
1430  NEXT
1440  FOR J = 1 TO FACT%
1450  PRINT FDES$(J)
1460  PRINT PDES%(J)
1470  NEXT
1480  FOR J = 1 TO ANUM%
1490  FOR K = 1 TO FACT%
1500  PRINT GDES%(J,K)
1510  NEXT
1520  NEXT
1530  PRINT CD$;"CLOSE ";NAME$
1540  PRINT CD$;"LOCK ";NAME$
1550 SFLG% = 0
1560 WNDW% = 4: GOSUB 280
1570  RETURN
1580  REM  * READ DATA BASE
1590 WNDW% = 1: GOSUB 280
1600  GOSUB 1290
1610 WNDW% = 4: GOSUB 280
1620 A$ = "TO RETRIEVE ANALYSIS F
     ROM DISC,"
1630  GOSUB 1140
1640  GOSUB 5440
1650  PRINT "*** RETRIEVING ";NAM
     E$"; ***"
1660 PSUM% = 0
1670  PRINT CD$;"UNLOCK ";NAME$
1680  PRINT CD$;"OPEN ";NAME$
1690  GOSUB 5440
1700  PRINT "*** RETRIEVING ";NAM
     E$;" ***"
1710  PRINT CD$;"READ ";NAME$
1720  INPUT ANUM%,FACT%,PHI%,PLOW
     %,GHI%,GLOW%
1730  FOR J = 1 TO ANUM%
1740 ADES$(J) = ""
1750  GET INPT$
1760  IF INPT$ = RET$ THEN 1790
1770 ADES$(J) = ADES$(J) + INPT$
1780  GOTO 1750
1790  INPUT ARES(J)
1800  NEXT
1810  FOR J = 1 TO FACT%
```

```
ANALYSIS:  HOME PURCHASE
 RESULT ALTERNATIVE
    0.0  HONG KONG
    0.0  KOWLOON
    0.0  NEW TERRITORY
  000.0  NEW LANTAU ISLAND


PRIORITY FACTOR
   80.0  COST
   60.0  LOCATION
   40.0  MORTGAGE
   35.0  SCHOOLS

TO CHANGE A FACTOR'S PRIORITY,
  CHANGE VALUE (ARROW KEYS TO START,
  ANY OTHER KEY TO STOP),MOVE TO NEXT
  VALUE <SPACE KEY>,OR FINISH <RETURN>
```

FIG 14

```
ANALYSIS:  HOME PURCHASE
 RESULT ALTERNATIVE
    0.0  HONG KONG
    0.0  KOWLOON
    0.0  NEW TERRITORY
 9000.0  NEW LANTAU ISLAND


PRIORITY FACTOR
   80.0  COST
   60.0  LOCATION
   40.0 ?BANK LOAN
   35.0  SCHOOLS

TO   CHANGE A FACTOR'S NAME,
PRESS <RETURN> FOR
MORTGAGE
OR ENTER NEW NAME.
```

FIG 15

```
ANALYSIS:  HOME PURCHASE
 RESULT ALTERNATIVE
  505.0  HONG KONG
    0.0  KOWLOON
    0.0  NEW TERRITORY
 9000.0  NEW LANTAU ISLAND


ALTERNATIVE: KOWLOON
  GRADE FACTOR
   30.0  COST
    5.0  LOCATION
    3.0  BANK LOAN
    3.0  SCHOOLS

TO RANK THIS ALTERNATIVE,
  CHANGE VALUE (ARROW KEYS TO START,
  ANY OTHER KEY TO STOP),MOVE TO NEXT
  VALUE <SPACE KEY>,OR FINISH <RETURN>
```

FIG 16

```
1820 FDES$(J) = ""
1830  GET INPT$
1840  IF INPT$ = RET$ THEN 1870
1850 FDES$(J) = FDES$(J) + INPT$
1860  GOTO 1830
1870  INPUT PDES$(J)
1880 PSUM% = PSUM% + PDES%(J)
1890  NEXT
1900  FOR J = 1 TO ANUM%
1910  FOR K = 1 TO FACT%
1920  INPUT GDES%(J,K)
1930  NEXT
1940  NEXT
1950  PRINT  CHR$ (1): REM  * NOM
     ON FIX
1960  PRINT RET$;CD$;"CLOSE ";NAM
     E$: REM  *NOMON FIX
1970  PRINT CD$;"LOCK ";NAME$
1980 WNDW% = 4: GOSUB 280
1990  GOSUB 450
2000  RETURN
2010  REM  * CHANGE A NAME *
2020 WNDW% = 4: GOSUB 280
2030  PRINT A$
2040  PRINT "PRESS <RETURN> FOR "

2050  PRINT ALPH$
2060  PRINT "OR ENTER NEW NAME.";

2070  VTAB VLOC%: HTAB 10
2080  PRINT " "; SPC( 27);" ";: HTAB
     9
2090  GOSUB 1020
2100  VTAB VLOC%: HTAB 9: PRINT "
     ";ALPH$
2110 SFLG% = 1
2120  RETURN
2130  REM  * CHANGE A VALUE *
2140  IF MFLG% <  > 1 THEN 2200
2150 WNDW% = 4: GOSUB 280
2160  PRINT A$
2170  PRINT "  CHANGE VALUE (ARRO
     W KEYS TO START,"
2180  PRINT "  ANY OTHER KEY TO S
     TOP),MOVE TO NEXT"
2190  PRINT "  VALUE <SPACE KEY>,
     OR FINISH <RETURN>";
2200  VTAB VLOC%: HTAB 1
2210  FLASH : PRINT " ";: NORMAL

2220 SPD% = 0:CFLG% = 0
2230 INPT% =  PEEK ( - 16384) - 1
     28: POKE  - 16368,0
2240  IF INPT% < 0 THEN 2330
2250  IF INPT% = EXEC% THEN 2390
2260  IF INPT% = SKIP% THEN 2400
2270 SFLG% = 1
2280 A% = 0
2290  IF INPT% = ICR% AND SPD% >
      - 1 THEN A% = 1
2300  IF INPT% = DEC% AND SPD% <
     1 THEN A% =  - 1
2310 SPD% = SPD% + A%
2320  IF A% = 0 THEN SPD% = 0
2330  IF NUMB + SPD% > HI% THEN S
     PD% = HI% - NUMB
2340  IF NUMB + SPD% < LOW% THEN
     SPD% = LOW% - NUMB
2350 NUMB = NUMB + SPD%
2360  GOSUB 180
2370 A% = 250: GOSUB 5490
2380  GOTO 2230
2390 CFLG% = 1
2400  VTAB VLOC%: HTAB 1: PRINT "
     "
2410  RETURN
2420  REM  * SET ALT.'S GRADES*
2430  IF FACT% < 1 THEN  RETURN
2440 MFLG% = 1:A$ = "TO RANK THIS
      ALTERNATIVE,"
2450 FCUR% = 1
2460 VFIR% = 9
2470  GOSUB 740
2480 VLOC% = VFIR% + 1
2490 NUMB = GDES%(ACUR%,FCUR%)
2500 VLOC% = VLOC% + 1
2510 HI% = GHI%
2520 LOW% = GLOW%
2530  GOSUB 2140
2540 MFLG% = 0
2550 GDES%(ACUR%,FCUR%) = NUMB
2560  IF CFLG% = 1 THEN 2610
2570 FCUR% = FCUR% + 1
2580  IF FCUR% > FACT% THEN 2450
2590  IF VLOC% > 19 THEN 2460
2600  GOTO 2490
```

```
2610  GOSUB 130
2620  RETURN
2630  REM  * ADD ALTERNATIVE *
2640 WNDW% = 3: GOSUB 280
2650 WNDW% = 4: GOSUB 280
2660  IF ANUM% < AMAX% THEN 2720
2670 A% = 2: GOSUB 5550
2680  PRINT "I'M SORRY,BUT YOU CA
     NNOT HAVE MORE"
2690  PRINT "  THAN ";AMAX%;" AL
     TERNATIVES."
2700  GOSUB 5510
2710  RETURN
2720 ANUM% = ANUM% + 1
2730 VLOC% = 2 + ANUM%
2740 ALPH$ = "ALTERNATIVE #" +  STR$
     (ANUM%)
2750 A$ = "TO SET THE NEW ALTERNA
     TIVE'S NAME,"
2760  GOSUB 2020
2770 ADES$(ANUM%) = ALPH$
2780  IF FACT% < 1 THEN  RETURN
2790 ACUR% = ANUM%
2800 A% = (GHI% + GLOW%) / 2
2810  FOR J = 1 TO FACT%
2820 GDES%(ACUR%,J) = A%
2830  NEXT
2840  GOSUB 2430
2850  RETURN
2860  REM  * DELETE ALT. *
2870  IF ANUM% > 2 THEN 2940
2880 WNDW% = 4: GOSUB 280
2890 A% = 2: GOSUB 280
2900  PRINT "I'M SORRY,BUT YOU CA
     NNOT HAVE LESS"
2910  PRINT "  THAN 2 ALERNATIVES
     ."
2920  GOSUB 5510
2930  RETURN
2940 A$ = "TO CHOOSE ALTERNATIVE
     TO BE DELETED,":: GOSUB 5600
2950  IF ACUR% < 1 THEN  RETURN
2960 WNDW% = 4: GOSUB 280
2970 A$ = "DELETE " + ADES$(ACUR%
     ) + " "
2980 VLOC% = VFIR% + 1
2990  GOSUB 5690
3000  IF INPT% <  > 1 THEN  RETURN
3010  GOSUB 5440
3020  IF ACUR% = ANUM% THEN 3100
3030  FOR J = ACUR% TO ANUM% - 1
3040 ADES$(J) = ADES$(J + 1)
3050 ARES(J) = ARES(J + 1)
3060  FOR K = 1 TO FACT%
3070 GDES%(J,K) = GDES%(J + 1,K)
3080  NEXT
3090  NEXT
3100 ANUM% = ANUM% - 1
3110  GOSUB 450
3120  RETURN
3130  REM *CHANGE ALT.NAME *
3140 A$ = "TO CHANGE ALTERNATIVE'
     S NAME,": GOSUB 5600
3150  IF ACUR% < 1 THEN  RETURN
3160 ALPH$ = ADES$(ACUR%)
3170 VLOC% = 2 + ACUR%
3180  GOSUB 2020
3190 ADES$(ACUR%) = ALPH$
3200  RETURN
3210  REM  * SET FACTOR'S GRADES*
3220 MFLG% = 1:A$ = "TO RANK ALTE
     RNATIVES FOR THIS FACTOR,"
3230 ACUR% = 1
3240  GOSUB 880
3250 VLOC% = VFIR% = 1
3260 NUMB = GDES%(ACUR%,FCUR%)
3270 VLOC% = VLOC% + 1
3280 HI% = GHI%
3290 LOW% = GLOW%
3300  GOSUB 2140
3320 GDES%(ACUR%,FCUR%) = NUMB
3330  IF CFLG% = 1 THEN 3370
3340 ACUR% = ACUR% + 1
3350  IF ACUR% > ANUM% THEN 3230
3360  GOTO 3260
3370  GOSUB 130
3380  RETURN
3390  REM  * ADD FACTOR *
3400 WNDW% = 4: GOSUB 280
3410  IF FACT%< FMAX% THEN 3470
3420 A% = 2: GOSUB 5550
3430  PRINT "I'M SORRY,BUT YOU CA
     NNOT HAVE MORE"
3440  PRINT "  THEN ";FMAX%;" FAC
     TORS."
3450  GOSUB 5510
3460  RETURN
3470 FCUR% = 1
3480  IF FACT% + 1 > FCUR% + FLNG
     % - 1 THEN FCUR% = FCUR% + F
     LNG%
3490  GOSUB 590
3500 FACT% = FACT% + 1
3510 VLOC% = VFIR% + 2 + FACT%
     FCUR%
3520 ALPH$ = "FACTOR #" +  STR$
     FACT%)
3530 A$ = "TO SET THE NEW FACTOR'
     S NAME,"
3540  GOSUB 2020
3550 FDES$(FACT%) = ALPH$
3560 PDES%(FACT%) = (PHI% + PLOW%
     ) / 2
3570 HI% = PHI%:LOW% = PLOW%
3580 NUMB = PDES%(FACT%)
3590 MFLG% = 1:A$ = "TO SET FACTO
     R'S PRORTITY,"
3600  GOSUB 2140
3610 PDES%(FACT%) = NUMB
3620 PSUM% = PSUM% + PDES%(FACT%)
3630  IF NUM% < 1 THEN  RETURN
3640 FCUR% = FACT%
```

```
3650 A% = (GHI% + GLOW%) / 2
3660  FOR J = 1 TO ANUM%
3670 GDES%(J,FCUR%) = A%
3680  NEXT
3690  GOSUB 3220
3700  RETURN
3710  REM * DELETE A FACTOR*
3720 WNDW% = 4: GOSUB 280
3730  IF FACT% > 1 THEN 3790
3740 A% = 2: GOSUB 5550
3750  PRINT "I'M SORRY, BUT YOU C
     ANNOT HAVE LESS"
3760  PRINT "   THAN 1 FACTOR."
3770  GOSUB 5510
3780  RETURN
3790 A$ = "TO DELETE A FACTOR,": GOSUB
     5850
3800  IF FCUR% < 1 THEN  RETURN
3810 A$ = "DELETE" + FDES$(FCUR%)
     + " - "
3820  GOSUB 5690
3830  IF INPT% <  > 1 THEN  RETURN

3840 PSUM% = PSUM% - PDES%(FCUR%)

3850  IF FCUR% = FACT% THEN 3930
3860  FOR J = FCUR% TO FACT% - 1
3870 FDES$(J) = FDES$(J + 1)
3880 PDES$(J) = PDES$(J + 1)
3890  FOR K = 1 TO ANUM%
3900 GDES%(K,J) = GDES%(K,J + 1)
3910  NEXT
3920  NEXT
3930 FACT% = FACT% - 1
3940  GOSUB 130
3950  RETURN
3960  REM  * CHANGE PRIORITY*
3970  IF FACT% < 1 THEN  RETURN
3980 MFLG% = 1:A$ = "TO CHANGE A
     FACTOR'S PRIORITY,"
3990 FCUR% = 1
4000 VFIR% = 9
4010  GOSUB 590
4020 VLOC% = VFIR% + 1
4030 NUMB = PDES%(FCUR%)
4040 HI% = PHI%:LOW% = PLOW%
4050 VLOC% = VLOC% + 1
4060  GOSUB 2140
4070 MFLG% = 0
4080 PSUM% = PSUM% - PDES%(FCUR%)
     + NUMB
4090 PDES%(FCUR%) = NUMB
4100  IF CFLG% = 1 THEN 4150
4110 FCUR% = FCUR% + 1
4120  IF FCUR% > FACT% THEN 3990
4130  IF VLOC% > 19 THEN 4000
4140  GOTO 4030
4150  GOSUB 130
4160  RETURN
4170  REM  * CHANGEE FACT NAME*
4180 A$ = "TOOO CHANGE A FACTOR'S
     NAME,": GOSUB 5850
4190  IF FCUR% < 1 THEN  RETURN
4200 ALPH$ = FDES$(FCUR%)
4210 VLOC% = 10 + FCUR%
4220  IF VLOC% > 19 THEN VLOC% =
     VLOC% -  - FLNG%
4230  GOSUB 2020
4240 FDES$(FCUR%) = ALPH$
4250  RETURN
4260  REM  * INITIALIZE DATA BASE
     *
4270 WNDW% = 1: GOSUB 280
4280  GOSUB 1290
4290 NAME$ = "TRADE STUDY"
4300 A$ = "TO START A NEW ANALYSI
     S,"
4310  GOSUB 1140
4320 WNDW% = 1: GOSUB 280
4330  PRINT "ANALYSIS:  ";NAME$
4340  PRINT "  VALUE PARAMETERS"
4350 AHI% = 2
4360 FHI% = 1
4370 PHI% = 100
4380 PLOW% = 1
4390 GHI% = 100
4400 GLOW% = 1
4410 AFLG% = 0
4420 NUMB = AHI%:VLOC% = 3: GOSUB
     180
4430  PRINT "  NUMBER OF ALTERNAT
     IVES"
4440 NUMB = FHI%:VLOC% = 4: GOSUB
     180
4450  PRINT "  NUMBER OF FACTORS
     "
4460 NUMB = PHI%:VLOC% = 5: GOSUB
     180
4470  PRINT "  HIGHEST FACTOR PRI
     ORITY"
4480 NUMB = PLOW%:VLOC% = 6: GOSUB
     180
4490  PRINT "  LOWEST FACTOR PRIO
     RIT       Y"
4500 NUMB = GHI%:VLOC% = 7: GOSUB
     180
4510  PRINT "  HIGHEST POSSIBLE S
     CORE"
4520 NUMB = GLOW%:VLOC% = 8: GOSUB
     180
4530  PRINT "  LOWEST POSSIBLE SC
     ORE"
4540 NUMB = AFLG%:VLOC% = 9: GOSUB
     180
4550  PRINT "  SET ALTERNATIVES F
     IRST(0)"
4560  PRINT  SPC( 7);"  OR  FACTO
     RS FIRST (1)";
4570 MFLG% = 1:A$ = "TO CHANGE A
     PARAMETER,"
4580 VLOC% = 3
4590 NUMB = AHI%:HI% = AMAX%:LOW%
     = 2
4600  GOSUB 2140
4610 AHI% = NUMB
4620 MFLG% = 0
```

```
4630  IF CFLG% = 1 THEN 4990
4640 VLOC% = VLOC% + 1
4650 NUMB = FHI%:HI% = FMAX%:LOW%
      = 1
4660  GOSUB 2140
4670 FHI% = NUMB
4680  IF CFLG% = 1 THEN 4990
4690 VLOC% = VLOC% + 1
4700 NUMB = PHI%:HI% = 100:LOW% =
     PLOW% + 1
4710  GOSUB 2140
4720 PHI% = NUMB
4730  IF CFLG% = 1 THEN 4990
4740 VLOC% = VLOC% + 1
4750 NUMB = PLOW%:HI% = PHI% - 1:
     LOW% = 1
4760  GOSUB 2140
4770 PLOW% = NUMB
4780  IF CFLG% = 1 THEN 4990
4790 VLOC% = VLOC% + 1
4800 NUMB = GHI%:HI% = 100:LOW% =
     GLOW% + 1
4810  GOSUB 2140
4820 GHI% = NUMB
4830  IF CFLG% = 1 THEN 4990
4840 VLOC% = VLOC% + 1
4850 NUMB = GLOW%:HI% = GHI% - 1:
     LOW% = 1
4860  GOSUB 2140
4870 GLOW% = NUMB
4880  IF CFLG% = 1 THEN 4990
4890 VLOC% = VLOC% + 1
4900 NUMB = AFLG%:HI% = 1:LOW% =
     0
4910  GOSUB 2140
4920 AFLG% = NUMB
4930  IF CFLG% = 1 THEN 4990
4940  GOTO 4570
4950 SFLG% = 1
4960 WNDW% = 1: GOSUB 280
4970  GOSUB 450
4980  GOSUB 130
4990 WNDW% = 1: GOSUB 280
5000 FACT% = 0:ANUM% = 0
5010  IF AFLG% < > 1 THEN 5100
5020  FOR J1 = 1 TO FHI%
5030  GOSUB 3400
5040  NEXT
5050  GOSUB 450
5060  FOR J1 = 1 TO AHI%
5070  GOSUB 2640
5080  NEXT
5090  GOTO 5170
5100  GOSUB 450
5110  FOR J1 = 1 TO AHI%
5120  GOSUB 2640
5130  NEXT
5140  FOR J1 = 1 TO FHI%
5150  GOSUB 3400
5160  NEXT
5170  RETURN
5180  REM  * HANDLE MENU *
5190 WNDW% = 4: GOSUB 280
5200  PRINT A$
5210  PRINT "  MOVE TO NEXT ITEM
     <SPACE BAR>,"
5220  PRINT "  SELECT ITEM <RETUR
     N>,OR EXIT <ESC>.";
5230 VCUR% = M1%
5240  VTAB VCUR%
5250  HTAB 2
5260  FLASH : PRINT " ";: NORMAL
5270  GOSUB 5800
5280  IF INPT% = CNCL% THEN INPT%
      = 0: GOTO 5390
5290  IF INPT% = EXEC% THEN INPT%
      = VCUR% - M1% + 1: GOTO 539
     0
5300  IF INPT% = SKIP% THEN 5330
5310 A% = 1: GOSUB 5550
5320  GOTO 5270
5330  VTAB VCUR%: HTAB 2
5340  PRINT " ";
5350 VCUR% = VCUR% + 1
5360  IF VCUR% <  = M2% THEN 5240
5370  IF PFLG% < > 1 THEN 5230
5380 INPT% =  - 1
5390 WNDW% = 4: GOSUB 280
5400  VTAB VCUR%: HTAB 2
5410  PRINT " ";
5420  RETURN
5430  REM  * PRINT BUSY ***
5440 WNDW% = 4
5450  GOSUB 280
5460  PRINT "*** PLEASE STAND BY
     ***"
5470  RETURN
5480  REM  * DELAY *
5490  IF A% > 0 THEN  FOR J = 1 TO
     A%: NEXT : RETURN
5510  PRINT
5520  INPUT "PRESS <RETURN> TO CO
     NTINUE.";INPT$
5530  RETURN
5540  REM  * PLAY A SOUND *
5550  IF A% < 1 OR A% > 4 THEN A%
      = 1
5560  FOR J = 1 TO A%: FOR K = 1 TO
      10
5570 X =  PEEK ( - 16336) +  PEEK
      ( - 16336)
5580  NEXT K: NEXT J: RETURN
5590  REM  * SELECT AN ALT. *
5600 VFIR% = 3
5610 M1% = VFIR%
5620 VBOT% = VFIR% + ANUM% - 1
5630 M2% = VBOT%
5640 PFLG% = 0
5650  GOSUB 5190
5660 ACUR% = INPT%
5670  RETURN
5680  REM  * CONFIRM CHICE ***
5690 WNDW% = 4: GOSUB 280
5700  PRINT A$
```

```
5710  PRINT "  EXECUTE <RETURN> O
     R CANCEL <ESC>?";
5720  FLASH : PRINT " ": NORMAL
5730  GOSUB 5800
5740  IF INPT% = EXEC% THEN INPT%
      = 1: GOTO 5770
5750  IF INPT% = CNCL% THEN INPT%
      = 0: GOTO 5770
5760  GOTO 5730
5770 WNDW% = 4: GOSUB 280
5780  RETURN
5790  REM    * GET KEYSTROKE ***
5800 INPT% =  PEEK ( - 16384) - 1
     28: POKE  - 16368,0
5810  IF INPT% >  - 1 THEN  RETURN

5820 A% = 100: GOSUB 5490
5830  GOTO 5800
5840  REM  * SELECT FACTOR *
5850 FCUR% = 1:PFLG% = 1
5860  GOSUB 590
5870 VLOC% = VFIR% + 2
5880 M1% = VLOC%
5890 M2% = VBOT%: IF M2% > M1% +
     FACT% - FCUR% THEN M2% = M1%
      + FACT% - FCUR%
5900  GOSUB 5190
5910  IF INPT% = 0 THEN FCUR% = 0
     : RETURN
5920  IF INPT% > 0 THEN FCUR% = I
     NPT% + FCUR% - 1: RETURN
5930 FCUR% = FCUR% + FLNG%
5940  IF FCUR% > FACT% THEN 5850
5950  GOTO 5860
5960  REM  * PRINT ANALYSIS *
5970 WNDW% = 4: GOSUB 280
5980  PRINT "ENTER RINTER SLOT NU
     MBER (1 TO 5 OR 7;"
5990  PRINT "  PRESS <RETURN> FOR
     ";SLOT%;")";
6000  INPUT "? ";INPT$
6010  IF  LEN (INPT$) = 0 THEN 60
     50
6020 INPT% =  VAL (INPT$)
6030  IF INPT% < 1 OR INPT% > 7 OR
     INPT% = 6 THEN 5970
6040 SLOT% = INPT%
6050 WNDW% = 4: GOSUB 280
6060  PRINT "PREPARE PRINTER FOR
     OUTPUT"
6070  GOSUB 5510
6080  GOSUB 5440
6090  PRINT "*** PRESS ANY KEY TO
      INTERRUPT ***"
6100 PAGE% = 0
6110  PRINT CD$;"PR#";SLOT%
6120  GOSUB 6440
6130 VLOC% = LINE%
6140  GOSUB 450
6150 LINE% = LINE% + 2 + ANUM%
6160  GOSUB 6540
6170  IF INPT% = 1 THEN 6380
6180  PRINT :LINE% = LINE% + 1
6190 VLOC% = LINE% + 2
6200 FCUR% = 1:TEMP% = FLNG%:FLNG
     % = FMAX%
6210  GOSUB 590
6220 FLNG% = TEMP%
6230  PRINT
6240 LINE% = LINE% + 3 + FACT%
6250 FCUR% = 0
6260  GOSUB 6540: IF INPT% = 1 THEN
     6380
6270 FCUR% = FCUR% + 1
6280  IF FCUR% > FACT% THEN 6380
6290  IF LINE% + 3 + ANUM% < 62 THEN
     6320
6300  GOSUB 6650
6310  GOSUB 6440
6320 ACUR% = 1
6330 VLOC% = LINE% + 2
6340  GOSUB 880
6350  PRINT
6360 LINE% = LINE% + 3 + ANUM%
6370  GOTO 6260
6380  GOSUB 6650
6390  PRINT CD$;"PR#0"
6400 WNDW% = 4: GOSUB 280
6410 PAGE% = 0
6420  RETURN
6430  REM  * PRIN HEADER *
6440  PRINT "."
6450  PRINT
6460 PAGE% = PAGE% + 1
6470  PRINT
6480  PRINT NAME$; SPC( 65 -  LEN
     (NAME$));"PAGE ";PAGE%
6490  PRINT
6500  PRINT
6510 LINE% = 7
6520  RETURN
6530  REM  * CHECK FOR LISTING AB
     OORT*
6540 INPT% =  PEEK ( - 16384) - 1
     28: POKE  - 16368,0
6550  IF INPT% < 0 THEN  RETURN
6560  PRINT CD$;"PR#0"
6570 A$ = "ABORT ANALYSIS LISTING
     -"
6580  GOSUB 5690
6590  IF INPT% = 1 THEN 6620
6600  GOSUB 5440
6610  PRINT "*** PRESS ANY KEY TO
      INTERRUPT ***"
6620  PRINT CD$;"PR#";SLOT%
6630  RETURN
6640  REM    * PRINT FOOTER ******

6650  IF LINE% > 0 THEN  FOR J =
     LINE% TO 66
6660  PRINT
6670  NEXT
6680  RETURN
6690  REM  * MAIN ROUTINE ******

6700  ONERR  GOTO 8430
```

```
6720  TEXT : HOME
6730  DATA   169,16,141,242,3,169
     ,3,141,243,3,73,165,141,244,
     3,96,32,234,3,76,18,212,104,
     168,104,166,223,154,72,152,7
     2,96
6740  FOR J = 768 TO 799
6750  READ K: POKE J,K
6760  NEXT
6770  CALL 768
6780  ONERR  GOTO 7650
6790 ERCNT = 0
6800  REM  * INTRODUCE PROGRAM *
6810  DIM GSUM%(5),GDES%(5,20),AR
     ES(5),ADES$(5),FDES$(20),PDE
     S%(20)
6820 WNDW% = 1: GOSUB 280
6830  PRINT "                EXECUTIVE
      ANALYZER"
6840  PRINT
6850  PRINT "               BY ALBE
     RT NG "
6860  PRINT "
          "
6870  PRINT "  COPYRIGHT 1984 BY
     COMPUTING AGE    "
6880  PRINT
6890  REM  * DECLARE VARABLES*
6900  REM  * DECLARE CONSTANTS *
6910 ALNG% = 5
6920 AMAX% = 5
6930 CD$ =  CHR$ (4)
6940 WLNG% = 25
6950 CNCL% = 27
6960 DEC% = 8
6970 EXEC% = 13
6980 FLNG% = 10
6990 FMAX% = 20
7000 ICR% = 21
7010 RET$ =  CHR$ (13)
7020 SKIP% = 32
7030  REM  * DECLARE VARIABES *
7040 SFLG% = 0
7050 NAME$ = "TRADE STUDY"
7060 PAGE% = 0
7070 SLOT% = 1
7080 BFLG% = 1
7090  REM  * HANDLE MAIN MENU*
7100 WNDW% = 3: GOSUB 280
7110 ERCNT = 0
7120 VBOT% = VFIR% + 1
7130  PRINT "    START A NEW ANALY
     SIS"
7140  PRINT "    RETRIEVE AN ANALY
     SIS FROM DISC"
7150  IF BFLG% = 1 THEN 7260
7160  PRINT "    ADD AN ALTERNATIV
     E"
7170  PRINT "    DELETE AN ALTERNA
     TIVE"
7180  PRINT "    CHANGE AN ALTERNA
     TIVE'S NAME"
7190  PRINT "    ADD A FACTOR"
7200  PRINT "    DELETE A FACTOR"
7210  PRINT "    CHANGE A FACTOR'S
     PRIORITY"
7220  PRINT "    CHANGE A FACTOR'S
     NAME"
7230  PRINT "    CHANGE AN ALTERNA
     TIVE'S GRADE"
7240  PRINT "   PRINT ANALYSIS";
7250 VBOT% = VFIR% + 10
7260 M1% = VFIR%:M2% = VBOT%
7270 A$ = "DO YOU WANT TO":PFLG% =
     0
7280  GOSUB 5190
7290  IF INPT% < 1 THEN 7580
7300 WNDW% = 3: GOSUB 280
7310  ON INPT% GOTO 7330,7350,737
     0,7390,7410,7430,7450,7470,7
     490,7510,7540
7320  GOTO 7100
7330  GOSUB 4270
7340  GOTO 7560
7350  GOSUB 1590
7360  GOTO 7560
7370  GOSUB 2640
7380  GOTO 7560
7390  GOSUB 2870
7400  GOTO 7560
7410  GOSUB 3140
7420  GOTO 7560
7430  GOSUB 3400
7440  GOTO 7560
7450  GOSUB 3720
7460  GOTO 7560
7470  GOSUB 3970
7480  GOTO 7560
7490  GOSUB 4180
7500  GOTO 7560
7510 A$ = "TO CHOOSE AN ALTERNATI
     VE TO BE CHANGED,": GOSUB 56
     00
7520  IF ACUR% > 0 THEN  GOSUB 24
     30
7530  GOTO 7560
7540  GOSUB 5970
7550  GOTO 7560
7560 BFLG% = 0
7570  GOTO 7100
7580  GOSUB 1290
7590 A$ = "EXIT PROGRAM -": GOSUB
     5690
7600  IF INPT% <  > 1 THEN 7100
7610 WNDW% = 1: GOSUB 280
7620  PRINT "BYE..."
7630  GOTO 8430
7640  REM  * ERROR PROCESSING **
7650  ONERR  GOTO 8310
7660  CALL 790
7670  IF PAGE% <  > 0 THEN  PRINT
      CD$;"PR#0"
7680 WNDW% = 4: GOSUB 280
7690 A% = 3: GOSUB 5550
7700  ON  PEEK (222) GOTO 8310,83
```

```
    10,8310,7740,8310,7800,8310,
    7880,7960,8030,8110,8310,811
    0
7710  IF  PEEK (222) = 56 OR  PEEK
    (222) = 255 THEN  HOME : CALL
     - 10621: GOTO 6820
7720  GOTO 8310
7730  REM  = WRITE-PROTECTED DISC
7740  FLASH : PRINT "WARNING";
7750  NORMAL : PRINT ":DISC HAS W
    RITE PROTECT"
7760  PRINT "TAB.REMOVE TAB OR GE
    T ANOTHER DISC."
7770  GOSUB 5510
7780  GOTO 8260
7790  REM  * FILE NOT FOUND
7800  REM
7810  PRINT "I CAN'T FIND THAT NA
    ME ON THIS DISC."
7820  PRINT "TRY A DIFFERENT NAME
     OR CHANGE DISCS."
7830  GOSUB 5510
7840 A$ = "TO CHANGE ANALYSIS NAM
    E,"
7850  GOSUB 1140
7860  GOTO 8260
7870  REM  * I/O ERROR
7880 ERCNT = ERCNT + 1
7890  IF ERCNT > 3 THEN 8310
7900  FLASH : PRINT "WARNING";
7910  NORMAL : PRINT ":DISC DRIVE
    R ERROR;VERIFY"
7920  PRINT "DISC IS INSERTED AND
     DOOR IS CLOSED."
7930  GOSUB 5510
7940  GOTO 8260
7950  REM  * FULL DISC
7960  PRINT CD$;"DELETE ";NAME$
7970  FLASH : PRINT "WARNING";
7980  NORMAL : PRINT ":  DISC IS
     FULL. PLEASE"
7990  PRINT "INSERT ANOTHER DISC.
    "
8000  GOSUB 5510
8010  GOTO 8260
8020  REM  * FILE LOCKED
8030  FLASH : PRINT "ARNING";
8040  NORMAL : PRINT T":FILE ALRE
    ADY EXISTS."
8050  INPUT "OK TO OVERWRITE (Y/N
    )?";INPT$
8060  IF INPT$ <  > "Y" AND INPT$
     <  > "N" THEN 8030
8070  IF INPT$ = "N" THEN A$ = "T
    O CHANGE ANALYSIS NAME,": GOSUB
    1140
8080  PRINT CD$;"UNLOCK ";NAME$
8090  GOTO 8260
8100  REM *FILE TYPE MISMATCH OR
    CTRL-C
8110  PRINT "THAT NAME IS NOT AA
    SOUND TABLE."
8120  PRINT "PLEASE TRY A DIFFERE
    NT NAME OR DISC."
8130  GOSUB 5510
8140 A$ = "TO CHANGE ANALYSIS NAM
    E,"
8150  GOSUB 1140
8160  GOTO 8260
8170  REM  * RESET OR CTRL-C STOP
8180  FLASH : PRINT "WARNING";
8190  NORMAL : PRINT ":   DO NOT
    PRESS CTRL-C OR"
8200  PRINT "RESET.  DO YOU WISH
     TO ABORT THE"
8210  INPUT "PROGRAM (Y/N)? ";INP
    T$
8220  IF INPT$ = "!" THEN  END
8230  IF INPT$ <  > "Y" AND INPT$
     <  > "N" THEN 8180
8240  IF INPT$ = "Y" THEN 8310
8250  GOTO 8260
8260 WNDW% = 4: GOSUB 280
8270  IF PAGE% <  > 0 THEN  PRINT
    CD$;"PR#";SLOT%
8280  ONERR  GOTO 7650
8290  RESUME
8300  REM  * NON-RECOVERABLE ERRO
    RS
8310  ONERR  GOTO 8410
8320  TEXT : HOME : IF  PEEK (222
    ) = 56 OR  PEEK (222) = 255 THEN
     CALL  - 10621: GOTO 6820
8330  FLASH : PRINT "WARNING";
8340  NORMAL : PRINT ":  PROGRAM
     HAS BEEN ABORTED"
8350  PRINT "DUE TO ONERR CODE ";
     PEEK (222);" ONE LINE"; PEEK
    (218) +  PEEK (219) * 256;".
    "
8360  PRINT
8370  PRINT "I WILL TRY TO SAVE Y
    OUR ANALYSIS"
8380  PRINT "UNDER THE NAME OF 'A
    DUMP'."
8390 NAME$ = "ADUMP"
8400  PRINT CD$;"UNLOCK ";NAME$
8410  ONERR  GOTO 8430
8420 SFLG% = 1: GOSUB 1290
8430  END
```

WORDSTAR高級應用篇：

蘋果長老

MAILMERGE 的操作備忘

很多朋友都會擁有MAILMERGE——一個與WORDSTAR配合使用的公用程式，但相信很多人都不懂得如何使用它，它本身的使用指南裏面介紹得很簡短，很多命令都沒有指出如何使用，因此我預備在這裏作一個MAILMERGE 的命令備忘錄，當各位參照過使用指南後對它的指令仍不甚了解時，參看本文會對各位有所帮助。

MAILMERGE是書信檔案和資料檔案合併印出的公用程式，它擁有着如下的特性：

(1)擴充文字處理系統的功能，
(2)它可以自動合併資料檔案和書信檔案來處理它們，
(3)它可以自動安排書信的規格，
(4)可以從一個記錄或多個的記錄之中讀取部份資料，
(5)能夠自動印出一系列的檔案，
(6)容許檔案中存在一些插入檔案，只以名稱出現。

MAILMERGE 控制指令

MAILMERGE 的控制命令大部份是點命令，藉以控制檔案間的關係和達到某些特殊的功能，它們最主要控制命令是

(1)．DF FILENAME；
這個命令用來設定包含這個點命令的書信檔案，它的資料檔名稱爲 FILENAME ，它與．RV 命令互相作用，由於資料檔案是由許多記錄所組成，一個．RV 對應一個記錄，．RV之中一個變數會對應記錄中的一個欄位。

．RV NAME1，NAME2，……NAMEN ：
這個命令可對應資料檔中的一個記錄，依照記錄中的欄位次序，逐一地讀入NAME1，NAME2，……NAMEN

(2)．AV"ASK FOR VARIABLE"，NAME，NO：
資料不再從資料檔讀入；在營幕上出現"ASK FOR VARIABLF"訊息，使用都依照其要求從鍵盤直接輸入資料，這些資料的前 NO 個字符讀入變數 NAME之中，若果不設定 NO 的話，則整行都讀入變數NAME之中，包括空句。

(3)．FI FILENAME：
若果書信檔之中包含此命令，就可以允許檔案的FILENAME 插入，當MAILMERGE執行到這個點命令時，執行權會轉到檔案FILENAME 上去，執行完後再回到書信檔中，此FILENAE 稱爲插入檔案，插入檔案本身可以擁有插入檔案，而MAILMERGE可以最高允許8個層次存在。

(4)．DM MESSAGE：
這個命令可以使MAILMERGE 在營幕上顯示MESSAGE 給使用者觀看。

在使用上，我先來討論如何自動合併資料檔和書信檔。

當同一內容的信件，要寄給相當數量而不同的客戶時，將書信建立成書信檔，其中帶有許多必要的點命令，以控制資料的讀取，客戶的資料建成資料檔，而一個客戶建成一個記錄，當使用文字處理系統的M時，自動把這兩個檔案合併，印出所要求的結果。

書信檔被重覆執行是由於檔案中存在．DF 命令，因此讀一個記錄，書信檔就重複一次。另外一種重複的方法是使用．RF ，如此可將．DF 放在引用檔案中(如有必要的話)。換句話說，包含有點命令．RP 的檔案將一直重複到資料結束，即使．DF 不在自己的檔案中。(．RP 須放在．RV 之後)，因此若果以．RP N來代替．RP的話，將重複執行包含此．RP N命令的檔案N次，直到資料用完。若果檔案中同時存在．DF 和．RP 時，．RP 沒有作用，因爲．DF 和．RV已經足夠使檔案重複執行，但如果沒有資料檔或，．RV與 ．RP 不存在同一檔案，則．RP 不會產生作用，亦即是說．RP 不會重覆執行。

由於篇幅所限，下期我將會把MAILMERGE如何控制表格及書信規格作一個介紹，而現在我以鍵入資料檔的注意事項來作完場：

(1) 鍵入以上所述的資料檔來使MAILMERGE執行之前，要以WORDSTAR 的N命令開啓非文件檔案而不是用D命令（開啓文件檔案）。
(2) 每個資料檔中的記錄須以記錄編號開始。
(3) 每個欄位以逗號隔開。
(4) 如果在欄位內包括了逗號，須以雙引號括起來。
(5) 其中每個欄位皆不能省略，可以一個空白來代替。
(6) 最後每個記錄以RETURN來結束。■

VISICALC的高級應用技術

令VISICALC變爲會思想的工具

陳耀南

VISICALC是電子報表程式的代表，但閣下可知道它是很容易被愚弄的嗎？以圖1這個生產力規劃報表爲例，假如你要得到第28週的可使用生產量（COLUMN F），而我們假設在F8（即COLUMN F，ROW8）之中工作天數是8，甚至—3時，VISICALC不會有甚麽反應，因爲我們並沒有告訴VISICALC工作天數不能大於7或不容許負數，而在F 17之中會計算出778 在工作天數爲8，或—292 在工作天數爲—3。

同樣地，若果你在F 11之中輸入20（假期的工作人數），VISICALC 亦不知道最多工作人數只有18人（F 10之中）而對這個20人作任何反應。

在以上這兩個典型問題，我們可總結得出不合理的數據輸入會令F 17（結果）產生無意義的數字，例如—61小時等。

圖2是用以產生圖1這個CAPACITY PLANNING 報表的程式列表。通常在很多現存的VISICALC報表之中都有著類似的格式和錯誤，不過VISICALC鮮爲人知的一項功能——核驗校對輸入數據，可以引用來不容許輸入無意義的數字（如每小時 工作 不容 許大於12（ROW 9）），又例如ROW 16裏的生產效率不容許不合理地增加至150％等（ROW 16）。

圖3是經過改進的CAPACITY PLANNING 的報表，它會把下列項目不合理數據拒絕輸入，經修改的地方是

① F8之中每週工作天數爲8，比7天更多

```
      A   B   C   D   E   F   G   H   I   J   K   L
 1  EXERCISE IN CAPACITY PLANNING:
 2  ==============================    DEPARTMENT:..........
 3
 4  PERIOD: DEC,1984                               TOTAL AV.
 5  ------------------------------------------     FOR   PER
 6  WEEK NO. ...............   28   29   30   31   32   MONTH WEEK
 7  ------------------------------------------------------------
 8  WORKING DAYS                5    5    5    4    5    24   4.80
 9  HOURS PER SHIFT             8    8    8    8    8    -    8.00
10  TOTAL NUMBER OF OPERATORS  18   18   19   19   20    -   18.80
11  NO. OF OPERATORS ON HOLIDAY2     2    1    1    0    -    1.20
12  AVAILABLE NO. OF OPERATORS 16   16   18   18   20    -   17.6
13  AVAILABLE WORKING HOURS   640  640  720  576  800  3376  675
14  % ALLOW FOR SICKNESS/LATNESS 5   4    3    3    2    -    3.33
15  EXPECTED ATTENDANCE HOURS 608  614  698  559  784  3264  653
16  EST.PRODUCTIVITY PERFORM (%)80  86   86   86   90    -    85.84
17  AVAIL.CAPACTY IN HOURS    486 528  601  480  706  2802   560
18  ==============================================================
19  CHECK ACCEPTABILITY OF DATA:
20  ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
21
22
23
24
```

圖1

② G11之中假期工作人數爲20，比G 10的總人數多

③ H16之中生產效率由30％增加至不合理的150％

圖3所使用的程式見於圖4，這種編寫技巧可以防止不合理的輸入數據，但凡：

① 輸入每週工作天數是負數或
每更工作小時是負數或
假期天工作人數是負數或
因病／事假／遲到人數是負數

② 輸入生產效率少於60％或
輸入因病／事假／遲到人數百分率高於25％

都被拒絕輸入計算。

各位要引用以上技巧到VISICALC的MODEL上時，必須對VISICALC 的邏輯函數多加注意，尤以@1F 函數爲然。同時亦要培養設計VISICALC公式，因爲這種技巧需要將一組公式藏在另一組之中，各位若果已掌握了基本VISICA—LC 使用後，爲何不致力於它的邏輯函數，你會從中得到很多收穫。不過有一點很遺憾，VISICALC的使用指南在這方面提供得很少，最主要的原因是它缺乏練習題目。

以下我將介紹一些很有用但很少人知的VISICALC函數，它們是 @1F，@NA，@MAX，@OR，@AND及@

```
      K8:  FORMAT    INTEGER       @SUM(F8...J8)
  L8:  FORMAT     2 DECIMAL        @AVERAGE(F8...J8)
  L9:  FORMAT     2 DECIMAL        @AVERAGE(F9...J9)
  L10: FORMAT     2 DECIMAL        @AVERAGE(F10...J10)
  L11: FORMAT     2 DECIMAL        @AVERAGR(F11...J11)
  F12:                             +F10-F11
  L12: FORMAT     2 DECIMAL        @AVERAGE(F12...J12)
  F13:                             +F12*F8*F9
  K13: FORMAT     INTEGER          @SUM(F13...J13)
  L13: FORMAT     INTEGER          @AVERAGE(F13...J13)
  L14: FORMAT     2 DECIMAL        +K13-K15/K13*100
  F15: FORMAT     INTEGER          100-F14*F13/100
  K15: FORMAT     INTEGER          @SUM(F15...J15)
  L15: FORMAT     INTEGER          @AVERAGE(F15...J15)
  L16: FORMAT     2 DECIMAL        +K17/K15*100
  F17: FORMAT     INTEGER          +F15*F16/100
  K17: FORMAT     INTEGER          @SUM(F17...J17)
  L17: FORMAT     INTEGER          @AVERAGE(F17...J17)
```

圖2

ISNA ，它們都被使用在圖4的程式之中。

首先參閱圖2之中的F12，它是＋F10－F11。爲何要將這公式改爲@1F（F11>＝0，@MAX（0，F10－F11），@NA）〔圖4中F12〕呢？要解答這個問題，我們先要看中間一段公式，它是MAX（0，F10－F11），以語言來解說它是我們要在F12中取兩個數之中較大的一個，若F10少於F11的話則取0。這公式的目的是確保F12不能存有負數，假如F10存有18而F11是20時，則結果會出現－2，但－2個工作人員是沒有意義的。而公式此時會檢查到－2，因爲－2少於0，因此F12會自動代入0，這種使用@MAX函數的技巧可以防止VISICALC的MODEL中不會出現負數。

F12之中的開頭及第三部份的公式設定@MAX函數計算工作，(0，F10－F11)應在F11大於或等於0時才起作用。

若F11是少於0的話，F12公式的第三部份會命令VIS－ICALC印出NA（NOT APPLICABLE的縮寫）到F12上去。同樣原理可以使用到F13，F15及F17之上的公式去，不過在ROW19之上的公式有時印出NA，有時却空白，它是如何做到的？秘密是ROW19的格式設定爲BAR CHA－RT（卽／F＊），若果結果正常的話應該是0，BAR CHART便呈空白，若果結果有若負數（卽不合理的結果）時，便會以NA回應。

從以上的一些探討中我們可以發覺到VISICALC是一個易學難精的工具，不過這種思考技術用在VISICALC身上確能令我們做出無懈可擊的SPREADSHEET。■

```
            A    B    C    D    E    F    G    H    I    J    K    L
 1 EXERCISE IN CAPACITY PLANNING:
 2 ==============================     DEPARTMENT:..........
 3
 4 PERIOD: DEC,1984                                        TOTAL AV.
 5 -------------------------------------------------       FOR   PER
 6 WEEK NO. ...............     28   29   30   31   32   MONTH WEEK
 7 ------------------------------------------------------------------
 8 WORKING DAYS                 8    5    5    4    5    27   5.40
 9 HOURS PER SHIFT              8    8    8    8    8    -    8.00
10 TOTAL NUMBER OF OPERATORS    18   18   19   19   20   -    18.8
11 NO. OF OPERATORS ON HOLIDAY  2    20   1    1    0    -    4.80
12 AVAILABLE NO. OF OPERATORS   16   0    18   18   20   -    14.40
13 AVAILABLE WORKING HOURS      na   0    720  576  800  na   na
14 % ALLOW FOR SICKNESS/LATNESS5     4    3    3    2    -    na
15 EXPECTED ATTENDANCE HOURS    na   0    698  559  784  na   na
16 EST.PRODUCTIVITY PERFORM (%)80    86   150  86   90   -    na
17 AVAIL.CAPACTY IN HOURS       na   0    1048 480  706  na   na
18 ==================================================================
19 CHECK ACCEPTABILITY OF DATA:           na             na   na
20 ++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
21
22
23
24
```

圖3

```
    K8:  FORMAT    INTEGER       @SUM(F8...J8)
L8:  FORMAT    2 DECIMAL     @AVERAGE(F8...J8)
L9:  FORMAT    2 DECIMAL     @AVERAGE(F9...J9)
L10: FORMAT    2 DECIMAL     @AVERAGE(F10...J10)
L11: FORMAT    2 DECIMAL     @AVERAGR(F11...J11)
F12:                         @IF(F11>=0,@MAX(0,F10-F11),@NA)
L12: FORMAT    2 DECIMAL     @AVERAGE(F12...J12)
F13:                         @IF(FB<=7,@MAX(0,F12*F8*F9),@NA)
K13: FORMAT    INTEGER       @SUM(F13...J13)
L13: FORMAT    INTEGER       @AVERAGE(F13...J13)
L14: FORMAT    2 DECIMAL     +K13-K15/K13*100
F15: FORMAT    INTEGER       @IF(F14<0,@NA,100-F14*F13/100)
K15: FORMAT    INTEGER       @SUM(F15...J15)
L15: FORMAT    INTEGER       @AVERAGE(F15...J15)
L16: FORMAT    2 DECIMAL     +K17/K15*100
F17: FORMAT    INTEGER       @IF(F16<0,@NA,F15*F16/100)
K17: FORMAT    INTEGER       @SUM(F17...J17)
L17: FORMAT    INTEGER       @AVERAGE(F17...J17)
F19: FORMAT    BAR CHART     IF(@AND(F19<=12,F14<=25,F16>60,F16<140),0,@NA)
K19: FORMAT    BAR CHART     @IF(@OR(@ISNA(F19),@ISNA(G19),@ISNA(H19),
                             @ISNA(I19),@ISNA(J19),@NA,0)
L19: FORMAT    BAR CHART     +K19
```

圖4

dBASE II PROGRAMMING進階篇

以標準格式編印 FINANCIAL STATEMENT

香港電腦研習中心供稿

上期我們擬定了一個測驗，測試你對DBASE Ⅱ在INTERACTIVE LEV—E L 運用的純熟程度。今期我們要把DBASE Ⅱ的運用，提高到PROGRAM—MING 的層次。

當你想進一步學習DBASE ⅡPRO—GRAMMING ，你會遇到一個困難，就是難以找到適當的參考書。市面上一DBASE Ⅱ 的參考書，詳於FUNDAM—ENTALS 的部份，而略於PROGR—AMMING。PROGRAMMING 部份的PROGRAM，不是太淺，便是太深。缺乏中等難度的PROGRAM，在一些淺白無味的例子後，是一整套的會計系統、郵遞系統之類，令人看得眼花繚亂，入手無門。

以下的一個PROGRAM ，是取自香港電腦研習中心PROGRAMMING IN DBASE Ⅱ 課程的教材，該課程包括由淺入深的各類大小SAMPLE PRO—GRAM 數十個，全部由本中心教職員編寫。初學者循序漸進，至最後可完全掌握DBASE Ⅱ 語言。

本程式名STATEMNT. CMD ，可產生標準格式的FINANCIAL STATE—MENT。我們將每一客戶的資料，包括客戶的名號、地址、結餘，存於CUS—TOMER. DBF 檔案中，結構如下：

```
STRUCTURE FOR FILE: CUSTOMER.DBF
NUMMBBER OF RECORDS:00003
DATE OF LAST UPDATE:00/00/00
PRIMARY USE DATABASE
FLD   NAME      TYPE  WIDTH  DEC
001   ACCOUNT    C     030
002   ADDR1      C     030
003   ADDR2      C     030
004   ADDR33     C     030
005   BALANCE    N     008    002
** TOTAL **           00129
```

```
                          KOWLOON BANK

                                       UNITED INTERNATIONAL LTD.,
                                       G. P. O. BOX 12345,
                                       HONG KONG.

------------------------------------------------------------------------
    DATE       CODE     DEBIT              CREDIT             BALANCE
------------------------------------------------------------------------

                                                            18600.00CR
  07/10/84     A13                        1200.00           19800.00CR
```

```
****************************************
* PROGRAM NAME: STATEMNT.CMD           *
* PROGRAMMER: STAFF OF HONG KONG       *
*             CENTRE OF COMPUTER       *
*             STUDIES (3-804040)       *
* DATE: 1ST DECEMBER 1984              *
* ALL COMMERCIAL RIGHTS RESERVED.      *
* THIS PROGRAM CAN BE USED FOR         *
* SELF-STUDYING PURPOSE SO LONG AS     *
* THE COPYRIGHT NOTICE IS ATTACHED.    *
****************************************
SET TALK OFF
SELECT PRIMARY
USE CUSTOMER INDEX CACCOUNT
SELECT SECONDARY
USE INTERACT INDEX IACCOUNT
SELECT PRIMARY
ERASE
?
? "            ***** PRINTING FINANCIAL STATEMENT *****"
?
? "PLEASE TURN ON PRINTER"
? "          AND POSITION PAPER"
?
WAIT
SET PRINT ON
SET FORMAT TO PRINT
SET CONSOLE OFF
STORE "----------" TO L
STORE L+L+L+L+L+L+L+L TO L
DO WHILE .NOT. EOF
     @ 2, 28 SAY CHR(14)+"KOWLOON BANK"
     @ 5, 45 SAY ACCOUNT
     @ 6, 45 SAY ADDR1
     @ 7, 45 SAY ADDR2
     @ 8, 45 SAY ADDR3
     ?
     ? L
     ? "          DATE        CODE      DEBIT              CREDIT;
        BALANCE"
     ? L
     ?
     STORE ACCOUNT TO KEY
     STORE BALANCE TO BL
     @ 13, 60 SAY BL
     IF BL>=0
          @ 13,70 SAY "CR"
     ELSE
```

```
                         KOWLOON BANK

                                      BINARY DATA PRODUCTS LTD.,
                                      2716, SUN HUNG KAI CENTRE,
                                      30, HARBOUR ROAD,
                                      WANCHAI, HONG KONG.
--------------------------------------------------------------------
   DATE      CODE    DEBIT                CREDIT        BALANCE
--------------------------------------------------------------------
                                                       7435.00CR
 07/05/84    A23     800.00                            6635.00CR
 07/09/84    A23                          750.00       7385.00CR
 07/14/84    A23     540.00                            6845.00CR
 07/20/84    A23                          400.00       7245.00CR
 07/20/84    A23                          300.00       7545.00CR
```

```
                @ 13,70 SAY "DR"
          ENDIF
          SELECT SECONDARY
          FIND &KEY
          STORE 1 TO C
          DO WHILE ACCOUNT = KEY .AND. (.NOT. EOF)
             IF PRINT
                 SKIP
             ELSE
                 STORE STR(C+13, 2) TO VT
                 @ &VT, 5 SAY DATE
                 @ &VT, 18 SAY CODE
                 IF DEBIT
                     @ &VT,25 SAY AMOUNT
                     STORE BL-AMOUNT TO BL
                 ELSE
                     @ &VT,45 SAY AMOUNT
                     STORE BL+AMOUNT TO BL
                 ENDIF
                 @ &VT,60 SAY BL
                 IF BL>=0
                     @ &VT,70 SAY "CR"
                 ELSE
                     @ &VT, 70 SAY "DR"
                 ENDIF
                 REPLACE PRINT WITH T
                 SKIP
                 STORE C+1 TO C
             ENDIF
          ENDDO
          SELECT PRIMARY
          REPLACE BALANCE WITH BL
          SKIP
          EJECT
       ENDDO
       SET PRINT OFF
       SET FORMAT TO SCREEN
       SET CONSOLE ON
       SET TALK ON
       SELECT PRIMARY
       USE
       SELECT SECONDARY
       USE
       RELEASE ALL
       RETURN
```

每一個客戶每月可能有零次至多次TRANSACTION，若我們將一次TR—ANSACTION 寄入一個RECORD中，每客戶可有多個RECORD 。我們將每一次TRANSACTION的資料記入INTERACT. DBF檔案中，結構如下：

```
STRUCTURE FOR  TEXT: INTERACT.DBF
NUMBER OF RECORDS:     00007
DATE OF LAST UPDATE: 00/00/00
PRIMARY USE DATABASE
FLD      NAME      TYPE WIDTH   DEC
001      ACCOUNT    C    030
002      DATE       C    008
003      AMOUNT     N    008
004      DEBIT      L    001
005      CODE       C    003
006      PRINT      L    001
** TOTAL **            00052
```

ACCOUNT 記顧客名號，DATE是交易日期，AMOUNT是銀碼，它可能是DEBIT或是CREDIT，若是DEB—IT，在DEBIT 欄內記入T或Y，若是CREDIT，則在DEBIT 欄內記入N或F。CODE是記交易類別的代號；PRINT 是記這次交易是否已在一次月結單中印出，若是，則下次無需再印。

程式動用了兩個DATABASE FILE，CUSTOMER. DBF 是按ACCOUNT排列，INTERACT.DBF是按ACCOU—NT及DATE 排列。在執行本程式前，要用INDEX ON ACCOUNT TO CACCOUNT及INDEX ON ACCOU—NT＋DATE TO IACCOUNT設立INDEX FILE。

工作的時候將CUSTOMER. DBF中的RECORD，逐個印出，每一個AC—COUNT印製一張STATEMENT，再到INTERACT.DBF 檔案中，取出所有屬於該ACCOUNT 的RECORD，逐一印出。每一次TRANSACTION佔用一行，CREDIT與DEBIT 的印刷位置不同，並計算及印出這次TRANS—ACTION 後的結存。印刷完後，把新的BALANCE 存回客戶記錄中BA—LANCE 項內。

程式中包括了大部份DBASE II指令及應用了很多DBASE II PROGRAM—MING 的技巧。但由於篇幅所限，未能爲讀者逐行解釋，唯有靠讀者自己領悟了。

祝早日修煉成功！■

40軌磁碟改造器

磁碟本身的內容在不受任何影響情況下INIT爲40軌

BABEL

相信不少讀者都擁有磁碟機，更可能不只一部的了。而現在市面上不少薄身磁碟機，都可以行40 TRACK DOS 的，加上現在已經發展了不少關於40 TRACK 的軟件，所以若不把磁碟格式改爲40 TRACK，實在有些浪費，由35 TRACK 改爲40 TRACK ，白白多了80個SECTOR，即是20K BYT—E 的容量。但問題却隨之而來了，但如果想把一隻已經存有程式的35 TRACK 磁碟，變爲40 TRACK 格式磁碟，該怎辦呢？辦法當然有的，你可以把這隻磁碟先COPY 一份，然後再INIT 爲40 TRACK，再用 F10 抄回那些程式便行了。但另一問題却隨之而來，如果那隻磁碟上的程式很多，又甚至需要變爲40 TRACK 的磁碟有很多，這樣做豈不是很花時間？兼且容易忙中有錯，所以筆者特爲此而寫了一個把35 TRACK 格式的磁碟，提升爲40 TRACK 格式，但又絲毫不損碟上其他內容。

原理概述

只要對APPLE DOS 3.3稍有研究的讀者，都知道在一DOS 3.3 格式下，磁碟的TRACK 17，SECTOR 0 是該磁碟的VTOC（冊內容表），這其實是一幅對映圖，標示出磁碟中 那些 SECTOR 已經使用，那些未用，亦記着該磁

```
25  REM  ::::::::::::::::::::::
30  REM  TRACK 35 TO TRACK 40
35  REM       BY  BABLE
40  REM  ::::::::::::::::::::::
50  PRINT "▬BLOAD RWTS"
100  HOME : TEXT : CLEAR
110  VTAB 1: HTAB 6: INVERSE : PRINT
     "APPLE ][ 40 TRACK UTILITY":
      NORMAL
120  VTAB 3: HTAB 2: PRINT "PROGR
     AM NAME:  INIT 35 TRACK TO 4
     0"
130  VTAB 5: HTAB 2: PRINT "BY:
     BABEL - COMPUHOUSE (C) 1984"
140  VTAB 7: HTAB 2: PRINT "DATE:
       17th NOVEMBER 1984"
150  VTAB 9: PRINT " ------------
     ---------------------------"
160  VTAB 12: HTAB 11: INVERSE : PRINT
     "SLOT   NO:"
170  VTAB 14: HTAB 11: PRINT "DRI
     VE  NO:"
180  VTAB 16: HTAB 11: PRINT "VOL
     UME NO:": NORMAL
190  VTAB 20: PRINT " -----------
     ----------------------------"
     : POKE 34,21
230  VTAB 12: HTAB 22: GET SL$: PRINT
     SL$:SL =  VAL (SL$)
240  IF SL < 1 OR SL > 7 THEN 230
250  VTAB 14: HTAB 22: GET DR$: PRINT
     DR$:DR =  VAL (DR$)
260  IF DR < 1 OR DR > 2 THEN 250
270  VTAB 23: HTAB 2: PRINT "PRES
     S Y FOR START, ANY KEY EXIT
     ";
280  GET Q$: IF Q$ <  > "Y" THEN
      TEXT : HOME : END
290  HOME
300 IOB = 3739
310  POKE (IOB + 1),(SL * 16): POKE
     (IOB + 2),DR
340  POKE (IOB + 15),(SL * 16): POKE
     (IOB + 16),DR
350  CALL 3584: GOSUB 1010
360 BF = 3840
370  IF  PEEK (BF + 52) < 36 THEN
     390
380  VTAB 23: HTAB 2: PRINT "THIS
      IS NOT A 35 TRACK DISK !!";
     : GET Q$: RUN 100
390  POKE (BF + 52),40
400 VL =  PEEK (BF + 6)
410  VTAB 16: HTAB 22: PRINT "
     ";: HTAB 22: PRINT VL
420  FOR A = 196 TO 212 STEP 4
430  POKE (BF + A),255: POKE (BF +
     A + 1),255
440  POKE (BF + A + 2),0: POKE (B
     F + A + 3),0
450  NEXT
460  CALL 3605: GOSUB 1010
470  CALL 3618: GOSUB 1010
500  IF  PEEK (3738) <  > 0 THEN
     1030: NEXT
510  VTAB 23: HTAB 2: PRINT "SUCC
     ESSFUL !! REMOVE THE DISK ";
     : GET Q$
520  HOME : VTAB 23: HTAB 2: PRINT
     "AGAIN ? ";: GET Q$
530  IF Q$ = "Y" THEN  RUN
540  TEXT : HOME : END
1010  IF  PEEK (IOB + 13) > 7 THEN
     1030
1020  RETURN
1030  VTAB 23: HTAB 2: PRINT "ERR
     OR OCCURED !!!";: GET Q$
1040  RUN 100
```

碟中的VOLUME NO. 及TRACK的數目。（讀者如果仍不明白，可以先參考本刊創刊號，P.73）要令磁碟知道自己有40條TRACK，便要改動這個表。但改好後，仍未能當 40 TRACK 使用的，因爲實際上第35－39TRACK 仍未被INIT，所以最後把第35－39TRACK INIT了，才能眞正成爲 40 TRACK 的磁碟。

使用方法

本程式是分爲BASIC 及機械程式各一段，BASIC 程式是因爲容易掌握，方便在畫面印出及簡單運算，主要還是機械語程式。讀者只要把BASIC 部份載入，然後RUN，程式自會再載入機械語言部份，然後執行。在程式運行時，如果程式發現VTOC 不能（或難於）讀出，或者在INIT 35－39TRACK 時，有錯誤發生的話，是會印出錯誤訊號的。如果讀者所放入的是40TRACK 磁碟的話，程式是會知道，而不予執行的。

後記

這個程式雖然把35TRACK 的磁碟改爲40TRACK，但不會改變該磁碟中的DOS 內容。程式在執行時，其中有一段時間只見磁碟機運行，而螢光幕却沒有任何反應，大概有二十多秒，這是程式在對35－39TRACK 作個別INIT。筆者原本是打算純以BASIC 來寫，只用POKE及CALL 指令來調用DOS 中的RWTS 便算了，但發覺用RWTS 來對35－39TRACK作個別INIT時，效果很差，所以便被逼再加一段機械副程式來作TRACK INIT 了。讀者不妨嘗試以純BASIC 或純用組合語言編寫，學習應用對DISK 的直接存取。■

本程式共有兩個檔案，其在程式磁碟上的檔案名是：

① T35 TO T40

② RWTS

```
  ::::::::
   RWTS
  ;:::::::
0E00- A9 11 8D 9F 0E EA EA EA
0E08- A9 01 8D A7 0E 20 91 0E
0E10- A9 00 85 48 60 A9 02 8D
0E18- A7 0E 20 91 0E A9 00 85
0E20- 48 60 A9 00 8D 9A 0E A9
0E28- 23 85 44 8D 9F 0E A9 00
0E30- 8D A7 0E 20 91 0E AE 99
0E38- 0E BD 89 C0 AD A9 0E 85
0E40- 41 A9 AA 85 3E A9 28 85
0E48- 45 A0 56 A9 00 99 FF BB
0E50- 88 D0 FA 99 00 BB 88 D0
0E58- FA 20 0D BF A9 08 B0 23
0E60- A9 30 8D 78 05 38 CE 78
0E68- 05 F0 18 20 44 B9 B0 F5
0E70- A5 2D D0 F1 20 DC B8 B0
0E78- 0A E6 44 A5 44 C9 28 D0
0E80- AA F0 03 8D 9A 0E AE 99
0E88- 0E BD 88 C0 A9 00 85 48
0E90- 60 A0 9B A9 0E 20 D9 03
0E98- 60 60 60 01 60 01 00 00
0EA0- 00 AC 0E 00 0F 00 00 00
0EA8- 00 00 60 01 00 01 EF D8
0EB0- A5 8A
```

REBOOT DOS

我想大家也可能嘗試過，當一看見一個精采的 BASIC程式，便急不及待地輸入電腦內，及至完成了大半，突然發覺忘記了 BOOT DOS，無法將程式記入磁碟內。如果現在才 BOOT DOS，就會失了已輸入的程式。改用錄音機會很繁復，而且容易出錯。現在介紹一個簡單的補救方法。

因爲在 BOOT DISK 過程中，會清洗部份 RAM，只要保留該段RAM的內容，便可以保存程式。當忘記 BOOT DOS，便要馬上進入MONITOR，將 BASIC 程式移往別處，然後 BOOT DOS，再將 BASIC 程式移放原地，最後用 SAVE記入磁碟內。

移動 BASIC 程式最重要知道程式的結尾在那裡，在地址裡 AF及 B0 便存入了程式的最後一句的地址。例如 AF及B0 分別是 24 1B，則程式是由 0801 至1B24。知道 BASIC 程式的起點及結束，便可以用 MONITOR 的MOVE 命令搬往別處了。

詳細步驟如下：

] CALL－151< RETURN >

*4100＜0800.5800M（移程式別處）

*4060＜60.FFM（保留原來程式的指標）

（此時插入 SYSTEM MASTER）

*C600G（BOOT DOS）

] CALL－151(進入MONITOR)

*60＜4060.40 FFM (回復原來指標)

*0800＜4100.9100M（將程式移返原處）

*CTRL－C< RETURN >

] SAVE APPLE（記入磁碟內，APPLE 是任意的檔案名）

現在，我相信你已把那精采的程式SAVE 入磁碟內了，再見。■

電腦時代

一本蘋果電腦用家必備雜誌

每期內容包括：

- 多個價值無可估計的軟件程式
- 專文深入討論週邊硬件的用途
- 最新的軟件介紹及評論
- 多個有深度而題材廣泛的專欄
- 還有其他精彩內容未能在此一一盡錄



第一期

COMPUTEXT 文字處理系統／電腦時代個人月曆／搬移 HELLO 騰空更多磁碟空間／重現被刪除之檔案／幼童數學四則運算練習／如何選購 AP II 電腦／上、下、左、右移動文字畫面／磁碟的結構／時日相距計算程式／過四關／增加英文字彙能力的遊戲／眞假隨機數………

第二期

彩色電視監視器色彩測試／沙蟹幻方局／貨櫃處理／美術字創造器／磁碟修護工具／數學套件程式／程式轉載器／如何用 F I D 抄錄有保護之磁碟／AP II 擴展插座之功能及應用探討／BAG OF TRICK 的拷貝辦法遊戲積分紀錄／抄錄程式的好幫手 WILDCARD ………



第三期

神經鬆弛體操／磁碟索引表封套製造器／雜誌目錄分類歸檔系統／DOS BOOT 碟的原理／動量不滅定律／幼童認數字初階／步步追踪／門外漢選購印字機／音樂產生的原理／蘋果編曲器／DOS 殺手／DOS 自動同步追踪器／圍棋遊戲／機械語言程式輸入方法／賽跑遊戲………



第四期

將 RAM 咭作爲磁碟機使用／太陽系行星移動探測系統／如何修補 DOS／立體迷宮／穿梭機特別任務／應用數學套件（二）／AP II 微電腦進化史／學前兒童幼稚園／觀看 TEXT FILE／DOS WRITER／電腦動畫製作原理／英文串字練習／PRODOS 操作系統介紹（上）／黑房計時器／失落的 STEP 及 TRACE ………





第五期

私人約會備忘日曆／立體空間動畫顯像系統／將 CP／M 檔案轉爲 DOS 檔案／80字行咭縱橫談／PRODOS 操作系統指令介紹（下）／6502 十三種存取資料地址模式／幼童數學練習數字咭／加數機大登場／神偷智破夾萬／隔山打牛／太空船降落任務／資料庫資料記錄原理／濃縮圖片儲存空間／蘋果萬花筒／DOS 指令簡化器／CP／M 專欄………

第六期

妙用無窮用的128K RAM 咭／APPLESOFT 擴展版本／私人會計系統／APPLESOFT 程式語言偵錯工具／DOS 緩衝器剖析／GO II／迷宮搜秘／太空侵略戰／亡命飛車／累加遊戲／神奇數學方塊／砌積木／機械人全面認識／CP／M 及 PASCAL 專欄／任意定義鍵盤／布倫時鐘／完整 CATALOG／爲磁碟命名………



訂閱收費表（包括郵費在內）

	本　港	澳門、中國、東南亞區	其　他　地　區	
雜誌加程式磁碟（十二期）	HK$455 ——	HK$500或US$65 HK$630或US$80	HK$530或US$68 HK$675或US$87	平　郵 空　郵
雜誌（十二期）	HK$96 ——	HK$118或US$15 HK$218或US$28	HK$140或US$18 HK$250或US$32	平　郵 空　郵







第七期　會說中文的蘋果／私人財政預算系統／私人財政收入系統／ PRODOS專欄／再談 80 字行咭／高解像圖轉載器／螢幕文本編輯處理器／蘋果杯大賽車（用 GRAPHICS MAGICIAN 編寫的遊戲）／高爾夫球遊戲／十一個 SECTOR／空戰驚魂／高解像字符繪劃器／ DOS3.3專欄…

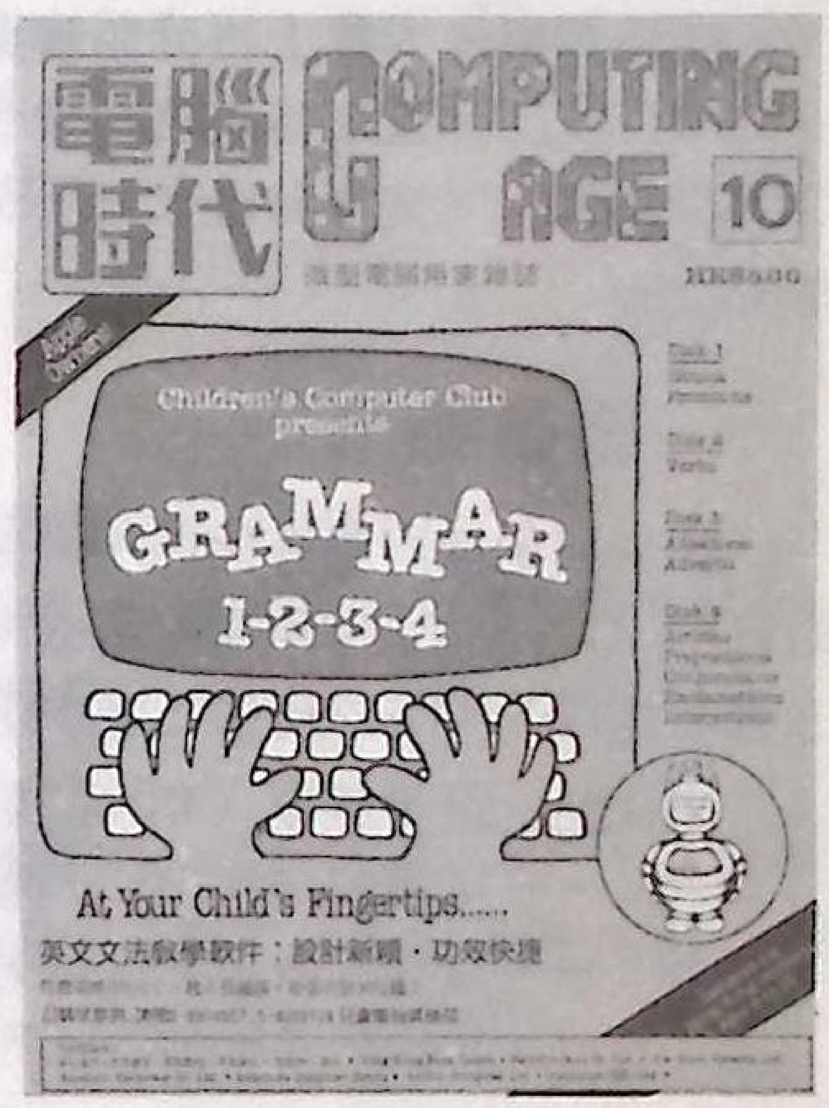
第八期　APPLESOFT滙編器／編印單個 LABEL／Z80 咭的認識／將 INTEGER 轉爲 APPLESOFT 方法／RESET 鍵特殊問題／高解像頁處理工具箱／眞正要動腦筋的小學四則運算／七海霸王（麥法基專欄）／原子黑洞／蛇之噩夢／有廿四種功能的 DOUBLE TAKE／自動三角函數曲綫繪畫系統／兩欄索引顯示………

第九期　磁碟檔案黃頁系統／增強鍵盤功能的 COMPUSOFT／混合機械語言 APPLESOFT 程式的技術初探／再談 Z80 咭地址配對方法／耐人研究的變數(上)／奇妙的&專欄／ APPLE TOOL KIT 各項優異功能／古堡之謎／七海霸王（下）／高空擲物／新 DOS 命令改換程式／捉迷藏／有趣的排序………

第十期　可隨意訂定格式之資料紀錄系統／ 16 K RAM 咭潛能發揮的竅門技巧／跳越加速器／ MACRO 鍵盤／機械語言入門／ 40磁軌工具程式之（一）／速度奇快的 TURBO PASCAL／ LISA 2・5 細談／將 DOS 移上 RAM 咭／培養兒童記憶能力／初中代數式運算／萬用數基轉換器／太空戰士（麥法基專欄）／射鴨仔／ 16 欄機械碼編印……

第11期　加速蘋果運作速度的多種方法探討／將 WORDSTAR 放入 RAMDIST使用（128 K RAM 咭應用篇）／魔域迷宮／VISICALC 專輯／家庭購物備忘及食物用品存貨紀錄系統／ PRINT SHOP 評介／電子辭典典範 SENSIBLE SPELLER／LOCK IT UP使用方法／JACKIES HEAD ／抄碟拆碟資料館／美化程式列表編印／英文語句文法分析練習系統／巨蟹求生記…

第12期　RAM 咭控制中心／家庭開支及個人消費管照系統／磁頭清潔器／諾曼第登陸戰／聲音和畫面技巧(麥法基專欄)／圖型建造師／符號逆輪計數機／電子郵箱／ LOCKSMITH 使用方法／ CP／M 55 個公用程式 POWER ／教育軟件評介專欄／ DTEST——測試你對 DBASE Ⅱ 的認識／加強 WORDSTAR 編印字體模式效果／有多項功能的 AMPERSAND 程式／工兵掘地雷／ E－Z DRAW 評介…







CAT GRAPHICS 指令簡介

55個新BASIC指令任意加入
普通程式中，提高繪畫功能

CHANG F.L.

"CAT GRAPHICS" 是最新推出的一個繪畫程式。這個軟件與其他繪畫程式不盡相同，因爲只要你LOAD入"CAT GRAPHICS"，你便可以有55個新BASIC 指令可供運用。它更可在PASCAL 下使用。你能夠在你自己普通程式中加入這些指令，令你程式的繪畫更加方便、美觀!!

這55個指令大致上可分爲繪畫指令（DRAWING COM—MANDS），圖形格式控制指令（GRAPHICS MODE CONTROL），文字幕指令（TEXT COMMANDS）及其他。

繪畫指令

在未開始探討指令之前，首先教大家如何才能在你自己的BASIC 程式上使用這些指令。在"CAT G"工作碟上有三個名爲GR1，GR2及GR5的FILE程式。

GR1——爲單解像（48K）版本（SINGLE RES.）

GR2——爲單解像（64K）版本（SINGLE RES.）

GR5——爲雙解像版本（DOUBLE RES.）〔只適用於Ⅱe〕

各位讀者只須依自己要求將上述其中一FILE 抄至自己要寫程式的磁碟上，用時則BRUN之，即可!!

注意，這些指令全是用& AMPERSAND 開頭的，而且當你LOAD 入上述程式時，程式是會被放置在DOS 之下的一段空白位置，所以是絕對不會對你寫程式時有任何影响或限制。此外，假如各指令中沒有特別提及，則此指令的用法無論是在單解像（APPLE Ⅱ＋）或雙解像版本（Ⅱe）都是一樣。

①& BOX

此指令之格式爲& BOX，X1，Y1，X2，Y2，它將畫出一個盒形物體，而其角座於座標X1，Y1及X2，Y2。

②& CIRCLE

格式：& CIRCLE，X，Y，RADIUS，X，Y爲圓心座標，RADIUS爲半徑之長度。

③& CURVE

格式：& CURVE，X1，Y1，X2，Y2，A，B， 其中A，B爲控制曲線圖形的數據，其值須爲－127至＋127。這樣就會畫一條CURVE 由X1，Y1至X2，Y2。

④& CLR

格式：& CLR，COLOR ，此指令係用你所指定的色彩去清除畫面。色彩選擇請參考表一。

⑤& COPY

格式：& COPY，MODE，功能在將一幅圖形從某一頁轉到其他情況下。有關情況則由MODE決定。

MODE 0～3 ：即將圖像由第一頁抄至第二頁

MODE 4～7 ：即將圖像由第二頁抄至第一頁

MODE 1及5 ：即將兩幅圖合併，不作任何刪除，原原本本地將所有圖／點顯示出來。

MODE 2及6 ：即將兩圖合併，但只保留兩圖重複之點。

MODE 3及7 ：將兩圖合併，但却刪去兩圖重複之點。

COLOR NO.	SINGLE RES. VERSION	DOUBLE RES. VERSION
0	BLACK 1	BLACK
1	GREEN	MAGENTA
2	PURPLE	DARK BLUE
3	WHITE 1	PURPLE
4	BLACK 2	DERK GREEN
5	ORANGE	GREY
6	BLUE	MEDIUM BLUE
7	WHITE 2	LIGHT BLUE
8	-	BROWN
9	-	ORANGE
10	-	GREY
11	-	PINK
12	-	GREEN
13	-	YELLOW
14	-	AQUE
15	-	WHITE

表一

*注意：在單解像版本中，色彩只有0～7可供選擇。

⑥& DOT

格式：& DOT，X，Y，在單解像下X之值須於0～279之間，而Y爲0～191間。但在雙解像下X則有不同，其值可爲0～139，0～279或0～559。而選擇是基於作者如何使用& RES 指令。而電腦則會於X，Y 畫出一點。〔& RES指令後文詳述〕

⑦& FILL

格式：& FILL，X，Y，COLOR ，其主要功能是在包含X，Y 點之地區塡上顏色。在單解像下，COLOR之值可由0～107，但在雙解像下，其值則由0～255。但有一點要注意的，若要在雙解像下執行此指令，必須首先SET好& TH—ICK的MODE。& THICK 乃一圖形格式控制指令。

⑧& FLIP

格式：& FLIP，COLOR，功能在改變畫面色彩，色彩選擇參考表一。

⑨& LINE

格式：& LINE，X1，Y1，X2，Y2 ，畫出一直線由X1，Y1至X2，Y2。

⑩& VSCROLL

格式：& VSCROLL，n， 功能在將畫面移上或移下n行，n 值須於－127至＋127之間，如n是負數，則代表畫面

CAT GRAPHICS 指令簡介

會向下移，正數則向上。在雙解像下，此指令只能於高解像度第一頁區工作，但單解像下，則無此限制。

⑪& HSCROLL

格式：& HSCROLL，n，功能類似& VSCROLL，A 是將畫面移左或右，n 則表BYTES之值（BYTES：GROU—PS OF 7 DOTS)。如n 是正數則移左，負數則移右。但在單解像下，1 BYTES＝7 DOTS，而雙解像下1 BYTES＝14 DOTS 。話雖如此，其移動位置相等。

⑫& PENGUIN

格式：& PENGUIN，X，Y，此指令功能較特別，主要是在X，Y 座標上畫出一隻企鵝。而由於是用造型繪製的，故在雙解像下，圖形會比較濶。

圖形格式控制指令

①& MODE

格式：& MODE，n（n係0～2其中一個）

0—畫一些東西在螢幕上

1—擦去螢幕上的東西

2—移動螢幕上的內容＊

＊當你選擇2時，係會將畫面上之內容刪去，但保留其背景。

②& COL

作用係指定圖形用什麼顏色畫出。格式：& COL，C—C 代表顏色。它和& FILL 指令一樣，顏色之選擇在單解像下係0～127，在雙解像下由0～255。

表二係色彩對照表，可供參考。

③& OUTLINE

此指令只對BOX及CIRCLE 發生作用。只要此指令被用，只有BOX及CIRCLE之外形（OUTLINE)會被畫出。

④& FULL

此指令作用剛與& OUTLINE相反，它會令到BOX 或CIRCLE被填滿。

⑤& SHAPEMODE

格式：& SHAPEMODE，n——n係0～7 其中一值，分別如下：

0—正常	4—反白
1—將濶度加倍	5—將濶度變成反白及加倍
2—將高度加倍	6—將高度加倍及變成反白
3—將高濶度都加倍	7—將高濶度都加倍及變成反白

此指令之工作情況在單及雙解像下都一樣。

⑥& THICK

此指令之工作，在單、雙解像下各有不同。

在單解像下，此指令會將用黑色或白色將字或其他東西變得較粗。

而在雙解像下，效果是比在單解像下粗一倍。

⑦& THIN

此指令之作用剛與& THICK 相反，在單解像下它令到

COLOR NO.	C.NAME	&COL 指令用之色彩參數
0	BLACK 1	80
1	GREEN	87
2	PURPLE	101
3	WHITE 1	77
4	BLACK 2	53
5	ORANGE	60
6	BLUE	70
7	WHITE 2	52

表(二)

用黑色或白色畫出來的東西變得非常幼細。

而在雙解像下，此指令允許你用黑色及白色作出560×192 的圖形。你可以選擇顏色0（黑色）或1（白色）。

⑧& SMOOTH

此指令係只工作於單解像下。格式：& SMOOTH，n—此指令作用在於使螢幕捲上時，變成又平滑又慢。n 之值越大時，則愈慢。而如果n＝0，則會恢復正常。

⑨& PAGE

格式：& PAGE，n——n 之值在單解像下是1或2，而在雙解像下n之值應是1。

作用在於選擇畫圖案於那一頁。在單解像下有2頁任由選擇。而雙解像下則只有一頁而已。

⑩& DISPLAY

格式：& DISPLAY，n——n之值是由1～10其中一數。具有顯示作用。

1—顯示高解像第一頁

2—顯示高解像第二頁

3—顯示高解像第一頁及文字窗（TEXT WINDOW)

4—顯示高解像第二頁及文字窗（TEXT WINDOW)

5—顯示低解像第一頁

6—顯示低解像第二頁

7—顯示低解像第一頁及文字窗

8—顯示低解像第二頁及文字窗

9—顯示文字幕第一頁

10—顯示文字幕第二頁

此指令在單或雙解像下之工作也略有不同。

如前所說，雙解像下，只有一頁高解像畫面可供使用。另外，文字幕及低解像畫面是顯示於80字行之形式下。如要在40字行下顯示，則要將n 值加16。

⑪& MAG

此指令使到你所畫的東西產生一個幻像（MAGNIFIED—VIEW）於其他頁之畫面上。

格式：& MAG，n，X，Y——n之值由0～3，X，Y是幻像之座標。

CAT GRAPHICS 指令簡介

0—刪去幻像　　　　2—產生4次幻像
1—產生兩次幻像　　3—產生8次幻像

注意：在雙解像下，要使用此指令，必須先設定& THI—CK之指令。

⑫& RESCAN

此指令是和& MAG 工作有關。如果用家並沒有使& MAG，則此指令是無作用的。

此指令是刪去所有在MAGNIFIED REGION的東西。

⑬& REFLECT

格式：& REFLECT，n， 此指令顯示畫面上的圖形從四個不同形式、途徑反射過來的情形。當n之值爲0時，表示關掉& REFLECT指令，當值爲1時則表打開& REFLE—CT 指令。

文字幕指令

①& UNHOOK

當你INITIALIZE"CAT GRAPHICS"後，所有的內容均會顯示在高解像畫面上。而此指令之作用便是DISCON—NECT文字及高解像畫面，即是說，使我們能夠在TEXT SCREEN 上顯示／印出內容。

②& REHOOK

此指令作用剛好與& UNHOOK 相反，當你使用& UN—HOOK指令，如你想在TEXT SCREEN 印出內容時，可用此指令。此指令可使你顯示內容於TEXT SCREEN及HI—RES PAGES。

③& AT

此指令能令你隨意設定CURSOR 之位置。格式爲：& AT，X，Y，而X，Y值之限制在單及雙解像下各有不同，詳細請參考& DOT指令。

④& LMARGIN

在一般情況下，當你打完一行程式，按〔RETURN〕後，CURSOR 會自動在左手邊出現。而此指令則可改變其位置，但只是改變X值（將SCREEN當爲座標圖來看），CURSOR 依然會在下一行出現，但不一定在最左，可在中間、最右。只要改變X值。

格式：& LMARGIN，X 。

⑤& HSPACE

格式：& HSPACE，n。如n＝1即代表1DOT

此指令是設定字與字之間的橫面空位。（參考圖1）

⑥& VSPACE

格式：& VSPACE，n 。

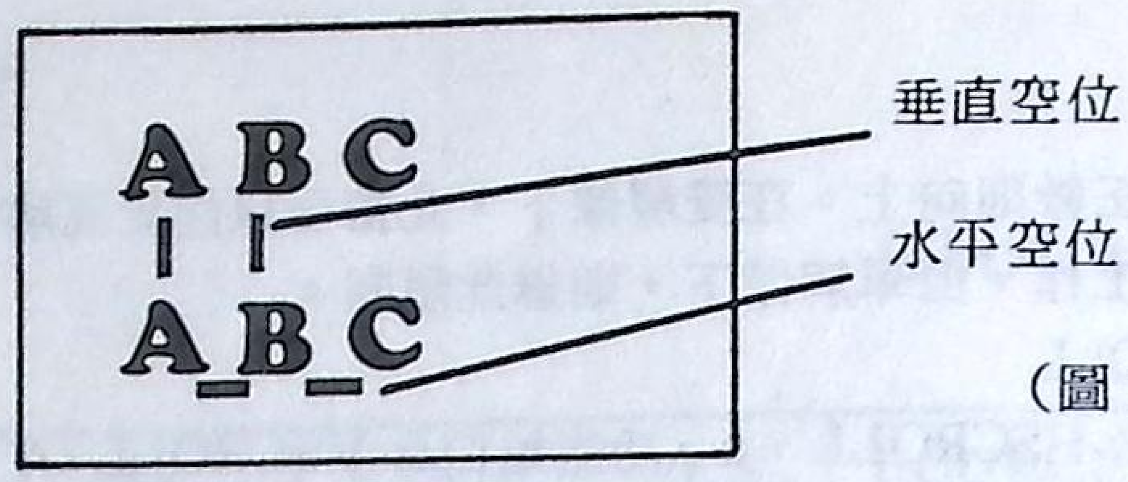
此指令和& HSPACE 相似，是設定字與字直切面的空位。（參考圖1）

⑦& REWRITE

此指令令到每一個字元之背景都變成黑色。在此指令下，你能在一些已有東西的畫面上寫／印出文字，並且是非常清楚的。

⑧& WRITE

此指令之功用是使你能重叠字元，於一個字上或於已設定顏色之背景上。



（圖1）

⑨& CHARMODE

格式：& CHARMODE，n。

此指令改變字元畫出之方式，n之值爲0～8。其分別如下：

0—正常顯示　　　　5—將寬度加倍及反白
1—將寬度加倍　　　6—將高度加倍及反白
2—將高度加倍　　　7—將高、寬度加倍及反白
3—將高、寬度加倍　8—（BOLD MODE）輪廓清楚顯示
4—反白顯示

＊注意：要使用BOLD MODE ，必須要在& WRITE MODE下。

⑩& TXTCOL

格式：& TXTCOL，C 。

此指令是設定字元／文字印出之顏色爲何。C之值在單解像下由0～107，而在雙解像下是由0～255 。

＊注意：在某些顏色下，除非你使用加大文字／字元印出之SIZE，否則，有時會顯示得不很清楚。

其他指令

①& LOAD

格式：& LOAD，n，"FILENAME"——"CAT GRAPHIC"裏有四種TABLE 可供使用：FONT TABLE、SHAPE TABLE、SOUND TABLE、BSHAPE TABLE。而這個指令便是將一個TABLE LOAD 入記憶系統。

n之值是由0～6，分別代表各類不同之TABLE。

1—FONT TABLE　　　4—BSHAPE TABLE
2—SHAPE TABLE　　 5—PACKED PICTURE
3—SOUND TABLE　　 6—BINARY FILE

②& ADDR

格式：& ADDR，VAR（VAR是塡入上一個二進碼檔案之起始位址）

此指令不單告訴你你所用& LOAD LOAD 入電腦的東西放在那裏，也告訴你二進碼檔案之起始點。

以下一個簡短程式容許你找出一個二進制檔案之起始點。

```
5  REM  -- BINARY START POINT FIN
     DER --
10  INPUT "NAME OF FILE?";A$
15  PRINT  CHR$ (4);"BLOAD";A$
20  & ADDR,X
25  PRINT "The binary start at ad
     dress";X;"
30  END
```

③＆STAB

格式：＆STAB，n

STAB指令作用是選擇第n個TABLE爲ACTIVE SH—APE TABLE。而由於第一個SHAPE TABLE 會自動LOAD入電腦而成爲ACTIVE的，故除非你想在同一時間有多幅SHAPE TABLE 可供使用的話，你毋須使用此指令。

④＆FTAB

格式：＆FTAB，n

此指令是選擇第n個FONT爲ACTIVE，但注意，由於ACTIVE FONT是已經被自動LOAD入，當你LOAD X "CAT GRAPHICS" 時，而你想加入另外一些FONTS，你必須將n之值設定爲2（由2開始）。

⑤＆SNTAB

作用：選擇nTH SOUND TABLE爲ACTIVE。

格式：＆SNTAB，n

⑥＆BTAB

格式：＆BTAB，n

功能爲選擇第n個BSHAPE TABLE爲ACTIVE。

但注意：此指令只能在單解像程式版本下使用。

⑦＆WIPEMEM

此指令作用是將所有TABLES洗去。（除了內置的第一個FONT。）

⑧＆START

此指令功能爲設定預留給TABLES用的起始位址。

格式：＆START，ADDN

⑨＆FINISH

格式：＆FINISH，ADDN

此指令功能在設定預留給TABLES用之終結位址。

⑩＆SOUND

格式：＆SOUND，n

此指令功能是PLAY出第n首音樂。

⑪＆NOISE

此乃另一個發聲指令，你能利用此指令產生一連串的聲音。

格式：＆NOISE，X1，X2，X3，X4，X5，X6，X7，X8，X9，X10，X11

X1是代表PITCH OF TONE1，而X3是代表PITCH OF TONE 3。X5是播出音樂多久（MAXIMUN：65535）。如果你將X2，X4，X6，X7，X8，X9，X10之值都設爲0，而X11爲1，而X1，X3，X5按需要設定其他值的話，則會產生一對音調偶音。如果你只想要單一個TONE，最好將X1，X3，X5之值設得相差不要太遠，例如只差1，2（即198,200）。

至於X2，X4是使PITCH在穩定頻率下改變。當你使用正數時，PITCH會較低，相反，如用負數，則PITCH 會較高。

而其他六個數之設定說明如下：

首先我要解釋爲何X11之值設爲1，因爲X11是告訴電腦要將X1—X5之音樂播出幾多次。如果X11爲1的話，則X2—X10之值都會被忽略。但是如果其值（X11）大過1，則當播完一次準備再重複的X5—X10 之音樂也會被加上。

故X6告訴電腦每一次PITCH之改變多少，而X7 則顯示第1個PITCH VALUE 之改變爲多少（RATE OF CHA—RGE OF FIRST PITCH VALUE），如此類推。

下面幾個例子將會產生很有趣的聲音，試試：

```
1  & NOISE,9,2,7, - 2,4000,4,0,0,
       - 777,6
2  & NOISE,111,30,7, - 19,65535, -
       43,12,11,0, - 777,1
3  & NOISE,30,1,7, - 1,65535,0,0,
       0,0,0,1
```

⑫＆CHKDOT

格式：＆CHKDOT，X，Y

此指令作用是在X，Y之座標上，用C 值所代表之顏色繪出X，Y 點。至於顏色方面可參照＆CLK指令及表一。

⑬＆PACK

格式：＆PACK，LOC，VAR

此指令是將現有的畫面壓縮而存入記憶系統。LOC 是你想將畫面存X的記憶位址之起始點，而VAR 則是此畫面用去之記憶空間長度值。

而當你存入後，其檔案名字會以．PIC結束。

用法如下：

(i)＆PEEK 792，N：IF N＞49000 THEN＆PEEK 103，N：N＝N＋2000——找出畫面所用之記憶空間。

(ii)接着：＆PACK，N，L

? CHR＄(4)；"BSARE"；N＄；"PAC.A"，N；"L"；L

注意：如果是要儲存雙解像畫面的話，則要將(ii)之"PAC"改爲"DPK"。

⑭＆UNPACK

格式：＆UNPACK，LOC

此指令是從記憶中取出一已儲存之畫面，放在現有的高解像畫面上。LOC是儲存我們需要之畫面的起始點。

⑮＆POKE

格式：＆POKE ADD，n

此指令與普通APPLESOFT BASIC 之POKE 指令不同。它會取n值之16BIT 值，放在ADD值及ADD＋1之記憶位址。那是ADD表記憶位置。

⑯＆PEEK

格式：＆PEEK ADD，VAR

此指令會將ADD及ADD＋1之值化爲16BIT 值，然後放在VAR。功能類似BASIC之PEEK 。

例如下列例子即是BASIC之KV＝PEEK 107＋256 * PEEK (1)：＆PEEK 0，KV

CAT GRAPHICS 指令簡介

⑰&PDL

格式：&PDL n，VAR

此指令是讀取PADDLE NO. n（0或1），再將結果存入VAR。此指令與BASIC 略有不同。由於BASIC 之PDL（n）之值只由0—255，但在高解像下橫面是250點。

下列程式能用以測試你的PADDLE。

10&PDL0，X：PRINT X：GOTO 10。

⑱&DRIVE

此指令是用以指明使用那一部磁碟機。

格式：&DRIVE，n

例如：&DRIVE，1 卽使用DRIVE 1

好了！CAT GRAPHICS 所提供了指令簡介，也到此爲止！最後希望你借CAT GRAPHICS 發揮你的繪畫天才。■

CAT GRAPHICS 指令一覽表

繪劃指令

1 &BOX 2 &CIRCLE 3 &CURVE
4 &CLR 5 &COPY 6 &DOT
7 &FILL 8 &FLIP 9 &LINE
10 &VSCROLL 11 &HSCROLL
12 &PENGUIN

文字幕指令

1 &UNHOOK 2 &REHOOK
3 &AT 4 &LMARGIN
5 &HSPACE 6 &VSPACE
7 &REWRITE 8 &WRITE
9 &CHARMODE 10 &TXTCOL

圖形格式控制指令

1 &MODE 2 &COL
3 &OUTLINE 4 &FULL
5 &SHAPEMODE 6 &THICK
7 &THIN 8 &SMOOTH
9 &PAGE 10 &DISPLAY
11 &MAG 12 &RESCAN
13 &REFLECT

其他指令

1 &LOAD 2 &ADDR
3 &STAB 3 &STAB
5 &SNTAB 6 &BTAB
7 &WIPEMEM 8 &START
9 &FINISH 10 &SOUND
11 &NOISE 12 &CHKDOT
13 &PACK 14 &UNPACK
15 &POKE 16 &PEEK
17 &PDL 18 &DRIVE

新游標

今次爲大家介紹一個非常有趣而且實用的程式——名爲"NEW CURSOR"。本來在蘋果電腦上之CURSOR（游標）是一個閃動的正方形，時間一久，令人覺得越來越枯竭無味。而本文提供了一個程式能夠改變閃動正方形之游標爲 他款色。例如：十、一、*、／、一…連英文字母也可以。

```
0300- 20 58 FF BA CA CA 9A 68
0308- 85 38 68 85 39 18 A9 1A
0310- 65 38 85 38 90 02 E6 39
0318- 20 EA 03 60 48 86 4E C9
0320- AB D0 02 A9 A0 48 A9 AB
0328- A4 24 D1 28 D0 02 68 48
0330- 91 28 A0 50 AD 00 C0 30
0338- 08 CA D0 F8 88 D0 F5 F0
0340- E5 68 A0 0A 20 E4 FB A4
0348- 24 A6 4E 68 4C 26 FD 00
0350- 00
```

而以下程式版本係將游標改爲閃動的"十"號。

現在首先教大家如何更改閃動的游標。在下列程式＄320 及＄327 之值爲"AB"，此乃閃動"十"之ASC 之值，故我們只需填入自己喜歡符號的ASC值於此二地址卽可。（注意：必須是16進制）

一個普通的字母或符號之ASC碼，我們可用？ASC( ) 測知，但注意，如果我們要的是閃動的字母或符號的話，必須將ASC碼＋128。此時只需我們將得出的ASC值轉爲16進制塡入＄320及＄327 卽可。

另外，此程式更有一項特點——游標移動時是有"CLICK！CLICK！"聲的。但如果用家不喜歡此聲，則可改變＄343 之值，例如：01、99、66……等都有不同的聲音。

如果閣下不喜歡有音樂的話，可輸入＄342：EA EA EA EA EA，則不會有任何聲响。

當閣下塡好自己需要的符號及聲音之值後，鍵入：BSAVE NEW-CURS—OR，A＄300，L＄4F卽可。■

CP/M 55個公用程式——POWER（下）

在第十一期中，筆者曾將POWER這套CP／M 工具程式套件其中的30項功能指令介紹給大家，相信各位經已清楚了解。今期筆者再將餘下的25個指令逐一詳細介紹出來，希望各位能從中吸取到有用的部份。

㉛SEARCH —將記憶體內的內容與所按入的ASC II 代碼，或十六進制碼或兩者混合的碼相比較，若果是符合，POWER 會將它們顯示出來，並將它們的所在位址一併印出，此項功能對程式維修或設計會很方便。

㉜SET —將磁碟上的檔案設定爲EXCLUSION或INCLUSION系統程式。

㉝SETDIR—將檔案設定在＄DIR之內，使檔案紀錄存在於DIRECTORY中。

㉞SETRO—將檔案設定爲＄R／O （只讀不可寫出狀態)。

㉟SETSYS—將檔案設定爲＄SYS內（不列入DIRECTORY之內）。

㊱SETWR—將檔案設定爲＄R／W(可讀及可寫出狀態)。

㊲SIZE —將檔案大小列出（包括SECTORS及KILO BYTE）。

㊳SORT—將DIRECTORY按字母排序。

㊴SPEED —設定螢幕上下捲動的速度。

㊵STAT—將磁碟上的磁碟使用空間及在剩餘空間列出。若果在中途更換過磁碟時按CTRL—C 可以更新這些內容。

㊶TEST—能夠測試磁碟介面是否可用作讀寫而又不會破壞原來已灌錄的資料，POWER 在這個過程之中會印出一個CHECKSUM數字，若果有BAD BLOCK，POWER 會將它們收集起來而設定爲略過狀態（BYPASS）。

㊷TYPE —將磁碟上的本文（TEXT）或ASCII 檔案傾印出來。

㊸TYPEA —將ASC II 檔案傾印出來，而輸出時加上行號及格式設定爲每行16個字元。

㊹TYPEH —將ＣＯＭ檔或其它BIN型檔傾印出來，而輸出格式和TYPEA功能相同。

㊺TYPEX —將ＣＯＭ檔或其它BIN型檔傾印出來，輸出格式設定行號，並每行16個字元，而且右邊將相對的ASC II字元印出。

㊻USER —將檔案設定爲EXCLUSIVE區域，由０至15。

㊼USRI —
㊽USR2 —
㊾USR3 —
㊿USR4 —
（這裏共有４個記憶區域給予使用者自定命令存放並執行之用。）

51 WRITE —將記憶體內容寫進任何磁碟上的任意磁軌及任意的磁區之中。

52 WRITEGR—將任何的CP／M檔案組寫入磁碟中。

53 XUSER —設定新的USER位址區域來接受檔案傳送之用，POWER提供了十五個可用區域來達成此項功能。

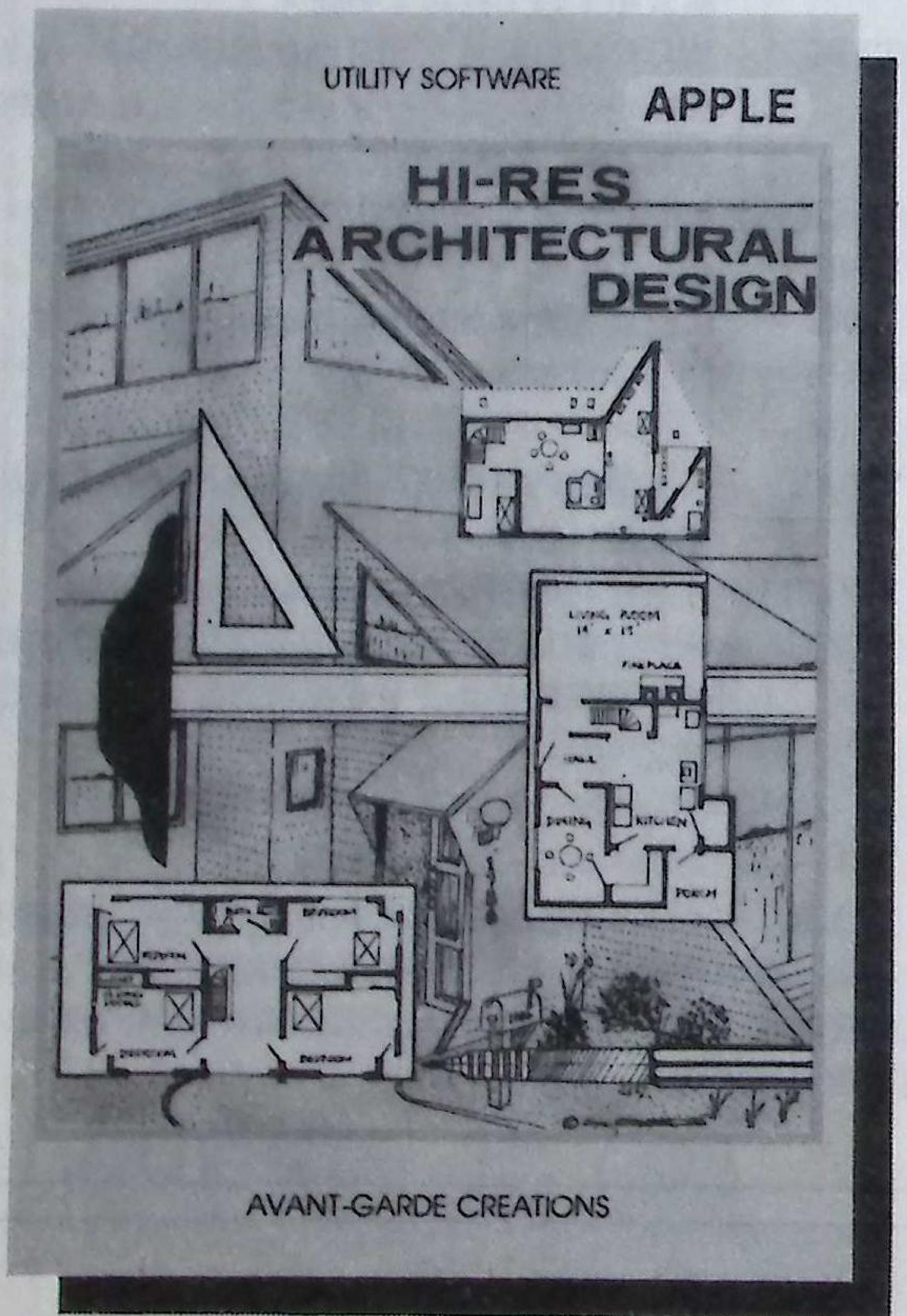
54 ？—將POWER的指令在螢幕上列出。

55 ？？—將額外加入的指令列出到螢幕上。

至此，POWER 所有的55個工具指令程式經全部爲大家介紹過。縱觀來說，這套工具對於我們日常使用CP／M 程式實在非常有用，它幫助我們減少處理瑣碎事項的麻煩。是值得爲大家推薦的。■

適合室內設計行業人士的軟件包

Hi-Res Architectural Design



軟件包外型

高解像室內設計工具包
名稱：HI－RES ARCHITECTURAL DESIGN
用途：室內設計
機種：APPLE Ⅱ 系列
出版公司：AVANT－GARDE CREATIONS
地址：P·O·BOX 30160 EUGENE，OR97403 USA
售價：$29·95美元

從事室內設計或建築公司的讀者相信對這個軟件包定會感到興趣。或者有意重新將舊居粉飾過年的朋友亦會對這件軟件覺得實用。

HI-RES ARCHITECTURAL DESIGN 能讓你在螢幕上劃出室內的平面圖，增刪任何的傢俬擺設，將平面圖印製硬本，在圖上加上細則說明文字，按比例將預先設計好的圖形（如餐桌、木門、冲涼缸等）放入室內平面圖上指定的位置。直到效果自己感到滿意爲止。

當將軟件BOOT 起後，將會有一個主索引供你選擇。假如是選擇繪畫模式，出現在螢幕最低部的將會是6個指令：

（1）UP／DOWN
（2）SIDEWAYS
（3）DIAGONAL
（4）ANGLE
（5）90－ANGLE
（6）SCALE

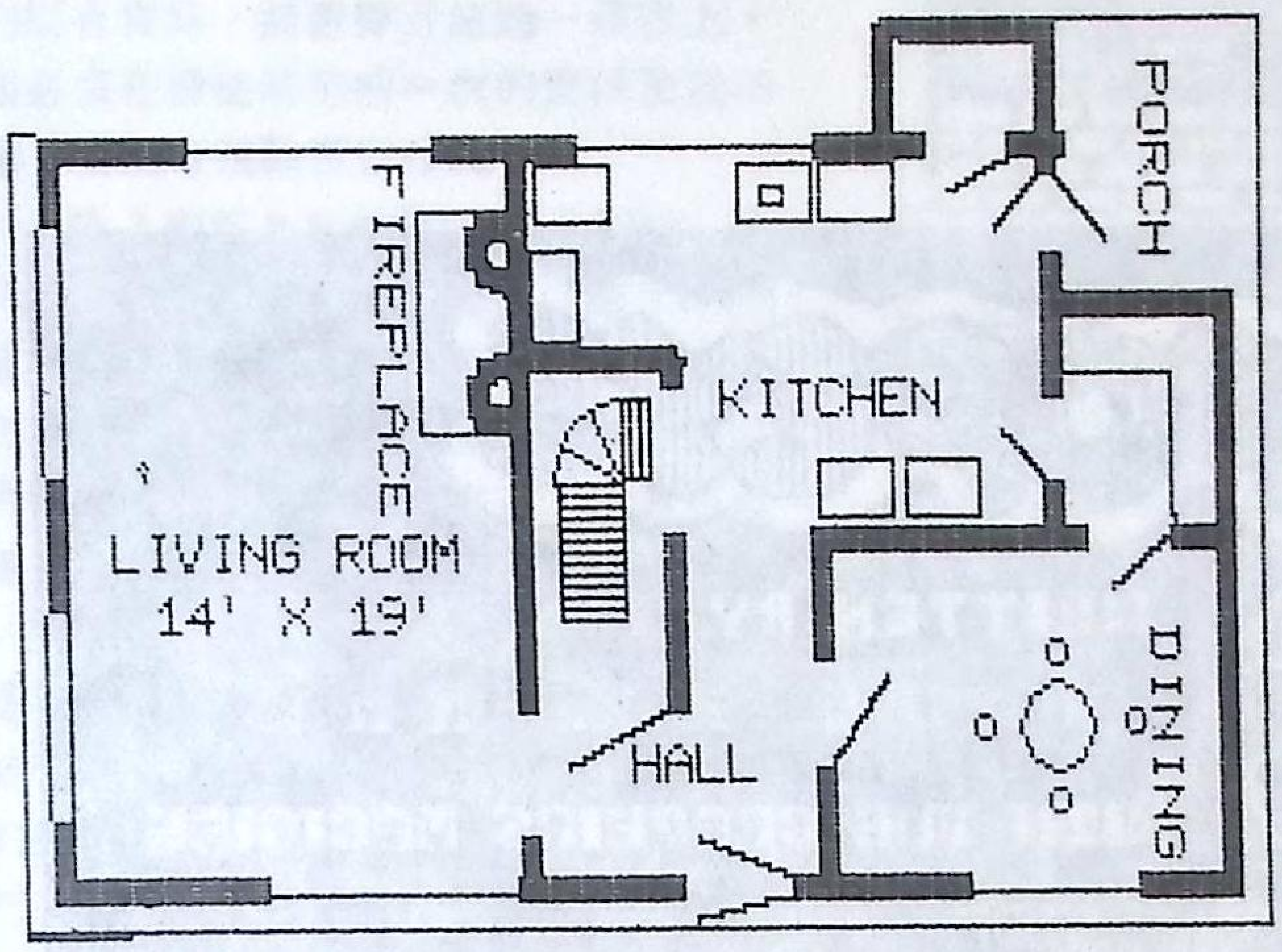
用家可用這些指令來作出初步的圖則。例如你要求的圖則面積比例是1比100，在螢幕上一條線的中間點則是50的單位比例（至於單位是呎是吋，則任由選擇）。當你指定兩點時，螢幕底部顯示出來的讀數將會指示你水平垂直、對角線或兩條線之間的角度。用家是利用控制桿或歡樂棒來控制繪畫。

要劃出直綫非常簡單，將游標設定在起始點（閣下可利用螢幕底部的讀數來決定準確的位置），按下（·）鍵，然後將游標移到端末點，再按同樣的鍵即可。直線有粗、幼、虛線等多種選擇（代表不同的牆壁）。更甚的是可以劃出彎角線，過程十分簡單。

本軟件一共有七十六個預繪圖形，包括一些經常要用的標準圖形及符號。用者祇要ESC，然後打入圖形的號碼。圖形將會在游標所停的位置上出現。游標位置可隨意設定。而且閣下可將圖形以90度的角度旋轉。因爲有些圖形是常用常見，故軟件的作者特別將這些圖形（總共26個）安排用單鍵來取用，例如B（BATHTUB）是浴缸，R是長方形（RECTANGLE）。有些圖形更可以作45°移動。

要擦除畫面的任何圖形，可用DRAW／╳DRAW。指令。要在高解像頁上劃線亦可做到。要消除大面積畫面，軟件分別有特別的符號來做。

適合室內設計行業人士的軟件包



利用本軟件繪畫出來的一幅平面圖（用CP－80 PRINTER印出）。

要在平面圖上加上文字說明一點也不困難。大家可選擇INSERT EXTRA LABELS 。你可任意用垂直或水平的英文文字說明，辦法是用PADDLE來將游標移至要加上說明的地方。不過，本軟件在這方面的效果未如理想，其中繪字的速度非常之慢。設計者不能即看到效果。同時，要準確定位並不是容易的事，這是本軟件美中不足的地方。呎（′），吋（″），度數（°）及公尺（*m*）的符號，十分齊備，供設計者使用。

這套軟件附有一本說明手册及指令表。由於程式相當易用，故此無須特別訓練。有關76個預置圖形，大家可在附同的參考咭上找到。

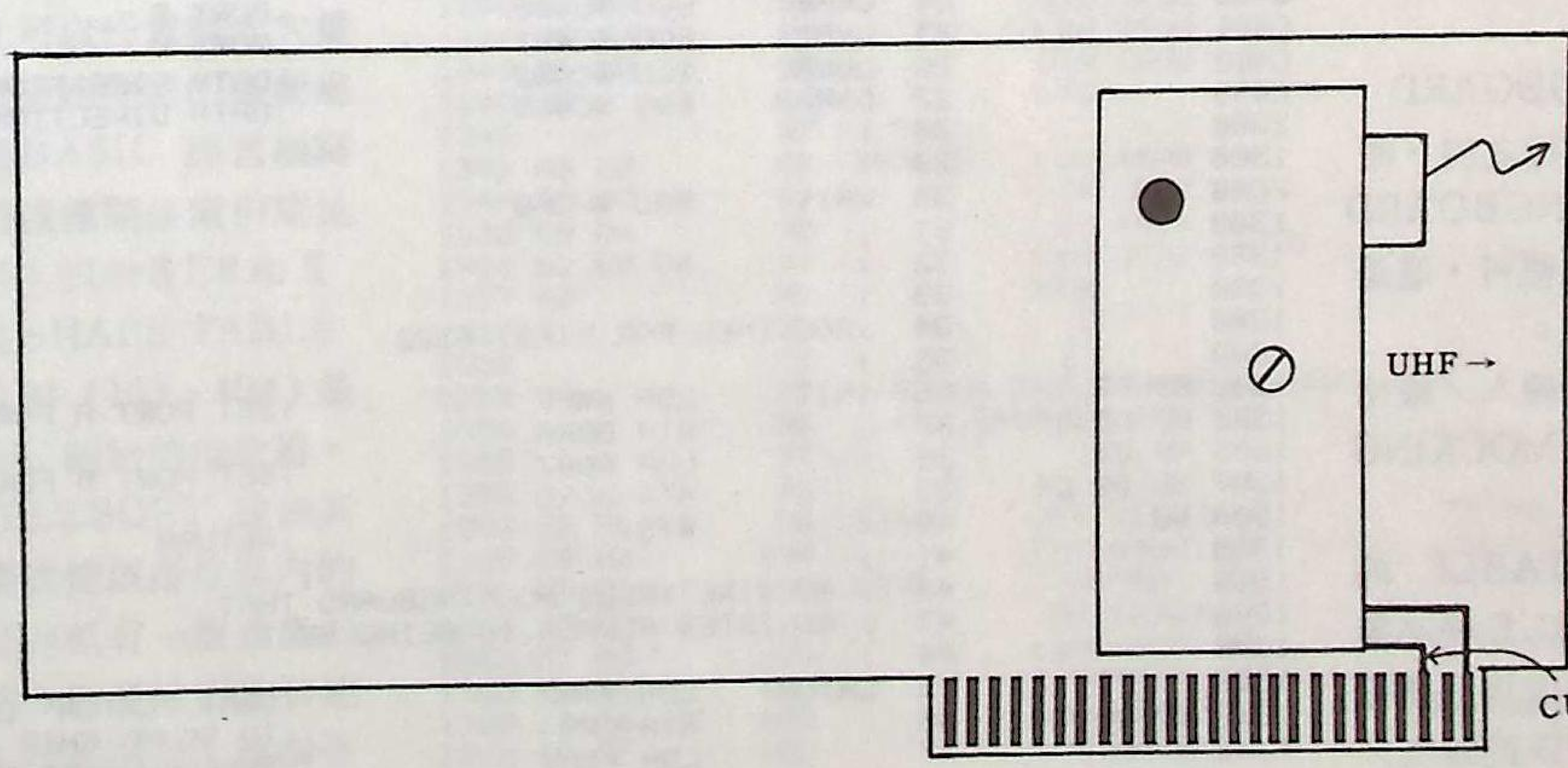
總括來說，這套軟件價錢十分相宜，用途亦適合一些專業人士。從事這行業的朋友，不妨購備一套。■

爲PAL CARD 加上顏色的文字顯示

莫信

正常的PAL CARD文字顯示，都是以黑底白字爲主，但若果你是用彩色電視機的話，只要將PAL CARD的綫路改動一下，便可以顯示不同的底色，例如：藍底白字、紅底白字等的變化，每種電視機所用的底色板有所不同，效果也有差異，不過只要調校PAL CARD上的色彩調校器，你就可以看到自己喜歡的顏色了（筆者建議讀者用綠底白字的顯示，因這可減少眼睛長時間對着顯示幕的疲勞），其實辦法很簡單，工具只是一把小刀，用來把PAL CARD面部的第三條接線切斷便成了（由主機尾部數起），若用過後不歡喜，也可用焊筆接回。（附：若讀者因改動PAL CARD綫路而引致電腦或PAL CARD的損壞，恕筆者不付責任。）

編者按：讀者如不想改動PAL任何部份，可用薄身膠紙貼上第三條接線，使不能接觸到SLOT的接點，效果亦會一樣。■



GRAPHICS MAGAICIAN編寫的遊戲：
大力水手POPEYE

麥法基

大力水手這個卡通人物大家都很熟識了，這一期的遊戲主角便是POPEYE，並且在下面談談如何利用MOCKING—BOARD 替程式配上精彩的聲音效果。

遊戲的規則是玩者操縱着POPEYE在畫面上四處接着跌下來的心形物體，不過要盡量避開布魯圖的追逐。POP—EYE 要總共接取二十粒心形物體，倘若超過九粒跌下海中的話則遊戲便完結了。在遊戲中有時會被利劍追着，按BUTTON 0 便可以把它們擊下，不過時間要十分配合，否則會被刺中。

雖然布魯圖是個強壯的人，但假如POPEYE 吃了大力菜後則可以把他擊倒。大力菜會不定時在畫面上出現。遊戲的開始時布魯圖並不曉得向POPEYE追逐，他只是隨意在畫面上亂打亂撞而已。但當難度提高時他會跟隨大力水手的路綫而追着，這時便要小心避開他了。

讀者如到電腦時代購買這個遊戲，在BOOT 起遊戲時電腦會詢問你是否裝設有MOCKINGBOARD，如果有的話，則按Y加以確定，電腦便會把MOCKINGBOARD 的程式LOAD 入，否則就由原來的APPLE 喇叭發聲，這點是要留意的。

現在請看看MOCKINGBOARD的SOURCE LISTING，共分爲數部份。由第 1 行至128 行是MOCKINGBOARD的主要程式，負責把聲音的資料交到卡上工作。現詳述如下：

INIT是觸發MOCKINGBOARD的程式，當任何程式開始運行之前，需要呼叫 INIT，否則MOCKINGBOARD不會發聲，但只要呼叫一次便可，這是開始所有輸入／輸出的控制。

LATCH是決定那一條輸入／輸出埠的應用，詳細資料請參閱MOCKING—BOARD 漫談。

WRITE 是把SOUND TABLE 的資料輸入，當SOUND TABLE 後必須把資料存放APPLE 記憶體內再由該程式轉送到MOCKINGBOARD 的輸入／




```
**END OF PASS 1

**END OF PASS 2

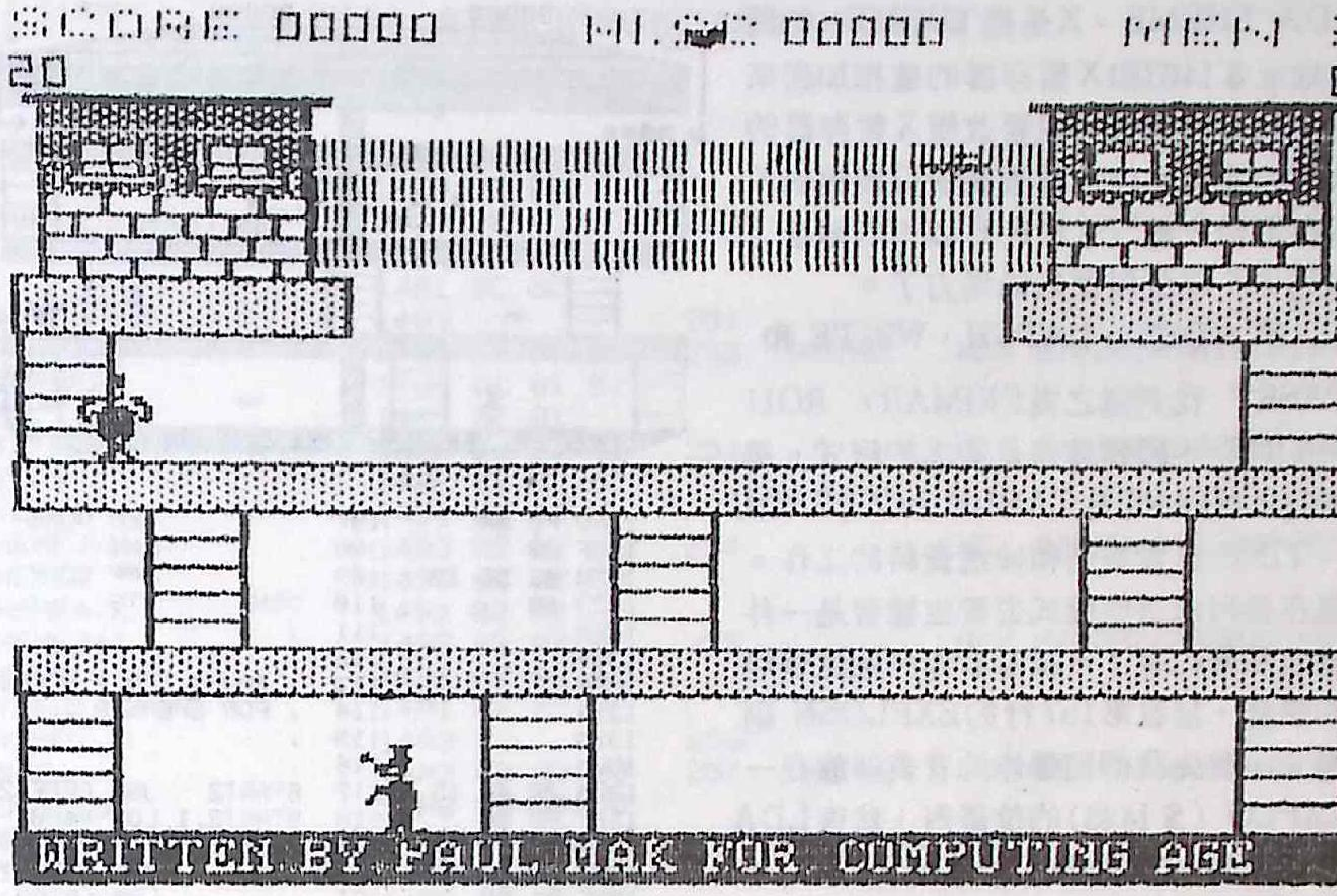
1300               1            ORG $1300
1300               2            OBJ $800
1300               3   ;
1300               4   ; ZERO PAGE ADDRESSES
1300               5   ;
1300               6   ;
0008               7   PTR      EPZ $08        ;SPEAKER 1 SOUND TABLE POINTER
000A               8   PTR2     EPZ $0A        ;SPEAKER 2 SOUND TABLE POINTER
00F9               9   TONE     EPZ $F9        ;LASER SOUND NOTES COUNTER
00EF              10   MTEMP2   EPZ $EF        ;BACKGROUND THEME COUNTER
1300              11   ;
1300              12   ;
1300              13   ; MOCKINGBOARD REGISTERS
1300              14   ; SPEAKER 1
1300              15   ;
C400              16   ORB      EQU $C400      ;PORT B
C401              17   ORA      EQU $C401      ;PORT A
C402              18   DDRB     EQU $C402      ;DATA DIRECTION REGISTER (B)
C403              19   DDRA     EQU $C403      ;DATA DIRECTION REGISTER (A)
1300              20   ;
1300              21   ; MOCKINGBORD REGISTER
1300              22   ; SEAKER 2
1300              23   ;
C480              24   ORB2     EQU $C480      ;PORT B
C481              25   ORA2     EQU $C481      ;PORT A
C482              26   DDRB2    EQU $C482      ;DATA DIRECTION REGISTER (B)
C483              27   DDRA2    EQU $C483      ;DATA DIRECTION REGISTER (A)
1300              28   ;
1300              29   ;
FCA8              30   WAIT     EQU $FCA8
1300              31   ;
1300              32   ;
1300              33   ;
1300              34   ;ROUTINES FOR FIRST 6522
1300              35   ;
1300 A9 FF        36   INIT     LDA #$FF       ;SET PORT A FOR OUTPUT
1302 8D 03 C4     37            STA DDRA
1305 A9 07        38            LDA #$07       ;SET PORT B FOR OUTPUT
1307 8D 02 C4     39            STA DDRB
130A 60           40            RTS            ;RETURN
130B              41   ;
130B              42   ; ROUTINE TELLS MOCKINGBOARD THAT
130B              43   ; REGISTER ADDRES IS BEING SENT
130B              44   ;
130B A9 07        45   LATCH    LDA #$07       ;SENT "LATCH" COMMAND
130D 8D 00 C4     46            STA ORB        ;TO SOUND CHIP
1310 A9 04        47            LDA #$04       ;THROUGH PORT B
```

輸出上。

RESET是清除MOCKINGBOARD的原有資料，倘若聲音超過一個以上，則必須在發聲時把前一次的資料加以清除，否則有殘餘聲音存在。

第7和第8行的PTR和PTR2 是SOUND TABLE的地址指位器，PTR是第一喇叭而PTR2是第二喇叭。筆者在MOCKINGBOARD漫談中曾詳細說明SOUND TABLE的創造，音表可以隨意存放在任何地址上，但MOCKING—BOARD需要知道記憶的位址。方法是利用間接索引地址模式去找尋，所以必須在零頁區內建立一對負責每邊喇叭的指位器，分別存放SOUND TABLE的低地址和高地址。讓我們看看第103行的指令LDA（PTR），Y，假設SOUND TABLE存放在＄1443 開始的位址，我們首先把低位址＄43存放在PTR 內，然後把高位址＄14存放在PTR＋1 內。程式第100 行把Y的索引暫存器設定爲零，LDA（PTR），Y這個指令便會從PTR和PTR＋1 中找出開始地址是＄1443，再把1443和Y的值相加起來，由於Y的值現在是零，所以加後的地址仍然是＄1443，最後把＄1443內的值（該值是＄00）由第104 行存放到ORA 的輸入／出埠內。取完第一個資料後並不完結，因爲SOUND TABLE每一種聲音均需要16個不同資料的數值，於是第106 行測試Y暫存器的值是否＄F，如果是的話則跳到110行完成整個提取過程。假如不是的話，（現在Y的值是零）所以把Y的值提升，第108 行的INY是把Y暫存器值加一，再跳到101 行繼續提取資料。

使用這種間接索引的方法去提取資料比較小限制，而且可以任意提取大量的資料，只要隨時改變指位器的值便造到了。通常就算使用BASIC 語言編寫程式亦很多時會應用這種間接索引定址法去找尋表格的資料。例如＄E8和＄E9（232，233）是SHAPE TABLE的指位器，＄67和＄68（103，104）是APPLESOFT BASIC開始的指位器。換言之假若想把APPLESOFT 改由其他地址開始，便需要改變該指位器內的值了。但APPLE 並非祇有一種查表的定址法，最通常用的一種是絕對索引地址模式，參看第174 行的指令；該指令




```
1312 8D 00 C4    48          STA ORB
1315 60          49          RTS                 ;RETURN
1316             50  ;
1316             51  ; TELLS MOCKINGBOARD THAT SOUND DATA
1316             52  ; IS BEING SENT
1316             53  ;
1316 A9 06       54  WRITE   LDA #$06            ;SEND "WRITE" COMMAND"
1318 8D 00 C4    55          STA ORB             ;TO SOUND CHIP
131B A9 04       56          LDA #$04            ;THROUGH PORT B
131D 8D 00 C4    57          STA ORB
1320 60          58          RTS                 ;RETURN
1321             59  ;
1321             60  ; STOP MOCKINGBOARD PRODUCING SOUND
1321             61  ;
1321 A9 00       62  RESET   LDA #$00
1323 8D 00 C4    63          STA ORB             ;SEND "RESET" COMMAND TO SOUND
 CHIP
1326 A9 04       64          LDA #$04            ;THROUGH PORT B
1328 8D 00 C4    65          STA ORB
132B 60          66          RTS                 ;RETURN
132C             67  ;
132C             68  ;ROUTINES FOR SECOND 6522
132C             69  ;
132C A9 FF       70  INIT2   LDA #$FF
132E 8D 83 C4    71          STA DDRA2
1331 A9 07       72          LDA #$07
1333 8D 82 C4    73          STA DDRB2
1336 60          74          RTS
1337             75  ;
1337 A9 07       76  LATCH2  LDA #$07
1339 8D 80 C4    77          STA ORB2
133C A9 04       78          LDA #$04
133E 8D 80 C4    79          STA ORB2
1341 60          80          RTS
1342             81  ;
1342 A9 06       82  WRITE2  LDA #$06
1344 8D 80 C4    83          STA ORB2
1347 A9 04       84          LDA #$04
1349 8D 80 C4    85          STA ORB2
134C 60          86          RTS
134D             87  ;
134D A9 00       88  RESET2  LDA #$00
134F 8D 80 C4    89          STA ORB2
1352 A9 04       90          LDA #$04
1354 8D 80 C4    91          STA ORB2
1357 60          92          RTS
1358             93  ;
1358             94  ;
1358             95  ; SOUND TABLE ACCESS ROUTINE
1358             96  ; FOR SPEAKER 1
1358             97  ;
1358             98  ;
1358 20 21 13    99  START   JSR RESET           ;CLEAR SOUND CHIP
135B A0 00      100          LDY #$00            ;USED TO IDENTIFY REGISTER
135D 8D 01 C4   101  LOOP    STA ORA             ;SOUND CHIP 1
1360 20 0B 13   102          JSR LATCH
1363 B1 08      103          LDA (PTR),Y         ;GET DATA FROM TABLE
1365 8D 01 C4   104          STA ORA
1368 20 16 13   105          JSR WRITE           ;STORE IN REGISTER
136B C0 0F      106          CPY #$0F            ;END OF DATA?
```

大力水手

LDA THEME，X是把THEME 的開始地址＄1463和X暫存器的值相加起來便成爲有效地址，只要改變X暫存器的值可以運用。但這種査表法只能提取X暫存器最大値（＄FF）以下的地址，超過255 種資料便無能爲力了。

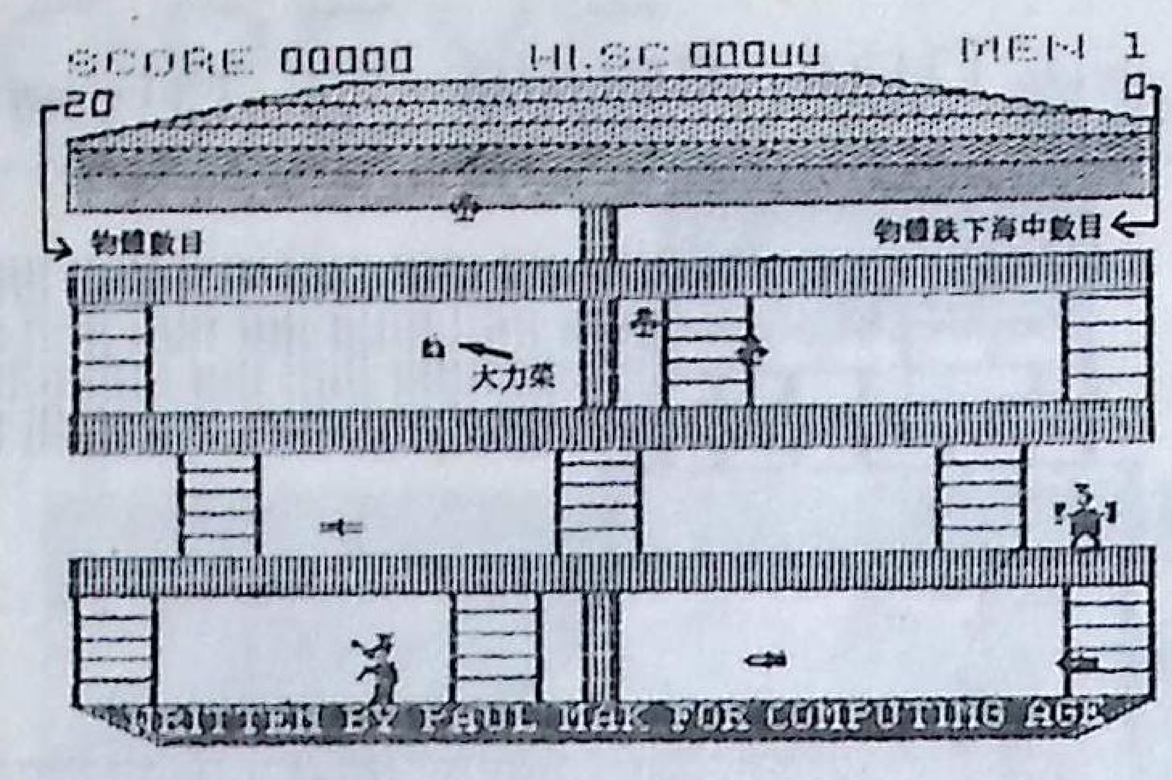
所有INIT，LATCH，WRITE 和RESET 我們稱之爲PRIMARY ROU—TINE ，因爲這些是基本的程式。第99行的START是 TABLE ACCESROU—TINE 負責査表和輸送資料的工作。現在要利用這些程式去產生聲音是一件輕而易舉的事了。試看看如何產生爆炸的聲音，參看第157行的EXPLOSN 副程式，首先我們把爆炸的音表存放在—EXPLO（＄1443)的位置內，然後LDA ＃EXPLO把低位址＄43存到PTR內，把／EXPLO 高位址存到PTR＋1 內，最後JSR START產生聲音，過程簡單而方便。應用時只要在遊戲內呼叫—EXPLOSN便可以。

在某些情況下使用MOCKINGBO—ARD 程式却要作出多少改變，參看第133 行的LASOUND 程式。當POP—EYE從高處跳下時會產生一種激光發射的聲音，我們需要以TONE作爲一個計數器，當TONE 的值越少，則聲音尖銳而當TONE的值加大時，聲音會漸漸便會低沉。TONE的值交由遊戲的主程式所控制，所以第146 行有一個RTS的指令。

再參看一下第167 行的MUSIC 程式，首先我們把一首曲的每個音符造成一個MUSIC TABLE 存在THEME和THEMEI內，因爲每一個音符由FINE TUNE和COARSE TUNE所决定，THEME是TUNE值的表而THEMEI是COARSE 值的表。POPEYE 的背景音樂總共有32個音符，所以第181 行的X暫存器測試是否＄20。

要留意的是在遊戲主程內必須有足夠時間讓音符發聲，因爲機械語言太快了，倘若每次運行都呼叫音樂程式必然造成沒有時間給音符發聲便被下一個音符的聲音所掩蓋，如此整首音樂會變成不知所謂。方法是在遊戲程式內設置一個時間的COUNTER ，讓主程式循環若干次才呼叫MUSIC ，這樣一首美妙的背景音樂便產生了。

下次再介紹另外一個遊戲，再會！



編者按：由於本遊戲的程式共有多個，這裏只能刊登有關利用MOCKING—BOARD 發聲的技巧部份。讀者要欣賞這個遊戲，可以購買即期的程式磁碟。

```
136D F0 04      107            BEQ DONE        ;YES,EXIT
136F C8         108            INY
1370 4C 5D 13   109            JMP LOOP        ;NO, GET NEXT DATA
1373 60         110  DONE      RTS             ;RETURN
1374            111  ;
1374            112  ;
1374            113  ; SOUND TABLE ACCESS ROUTINE
1374            114  ; FOR SPEAKER 2
1374            115  ;
1374            116  ;
1374 20 4D 13   117  START2    JSR RESET2
1377 A0 00      118  START2.1  LDY #$00
1379 8C 81 C4   119  LOOP2     STY ORA2
137C 20 37 13   120            JSR LATCH2
137F B1 0A      121            LDA (PTR2),Y
1381 8D 81 C4   122            STA ORA2
1384 20 42 13   123            JSR WRITE2
1387 C0 0F      124            CPY #$0F
1389 F0 04      125            BEQ DONE2
138B C8         126            INY
138C 4C 79 13   127            JMP LOOP2
138F 60         128  DONE2     RTS
1390            129  ;
1390            130  ; LASER SOUND ROUTINE
1390            131  ;
1390            132  ;
1390 A9 00      133  LASOUND   LDA #$00        ;LOAD HIGHEST
1392 85 F9      134            STA TONE        ;FREQUENCY VALUE
1394 A9 53      135            LDA #LTABLE     ;SET SOUND TABLE ADDRESS
1396 85 08      136            STA PTR         ;OF LOW ORDER BYTE
1398 A9 14      137            LDA /LTABLE     ;SET SOUND TABLE ADDRESS
139A 85 09      138            STA PTR+1       ;OF HIGHT ORDER BYTE
139C 20 58 13   139            JSR START       ;TRANSFER DATA
139F A9 00      140            LDA #$00        ;LATCH FIRST REGISTER
13A1 8D 01 C4   141            STA ORA         ;ADDRESS
13A4 20 0B 13   142            JSR LATCH
13A7 A5 F9      143  LA1.1     LDA TONE        ;GET TONE VALUE
13A9 8D 01 C4   144            STA ORA         ;STORE IN REGISTER
13AC 20 16 13   145            JSR WRITE
13AF 60         146            RTS             ;RETURN
13B0            147  ;
13B0 A9 00      148  LA1.2     LDA #$00        ;CLEAR SOUND CHIP
13B2 85 F9      149            STA TONE
13B4 20 21 13   150            JSR RESET
13B7 60         151            RTS
13B8            152  ;
13B8            153  ;
13B8            154  ; EXPLOSION SOUND ROUTINE
13B8            155  ;
13B8            156  ;
13B8 A9 43      157  EXPLOSN   LDA #EXPLO      ;SET TABLE ADDRESS
13BA 85 08      158            STA PTR         ;TO LOW ORDER BYTE POINTER
13BC A9 14      159            LDA /EXPLO      ;SET TABLE ADDRESS
13BE 85 09      160            STA PTR+1       ;TO HIGH ORDER BYTE POINTER
13C0 20 58 13   161            JSR START       ;TRANSFER DATA
13C3 60         162            RTS             ;RETURN
13C4            163  ;
13C4            164  ; BACKGROUND MUSIC
13C4            165  ;
13C4            166  ;
13C4 A9 00      167  MUSIC     LDA #$00        ;SET COUNTER TO 0
13C6 8D 17 15   168            STA MTEMP       ;TO MUSIC COUNTER
13C9 A9 A3      169  MUSIC1    LDA #THEME2     ;SET MUSIC TABLE ADDRESS
13CB 85 08      170            STA PTR         ;TO LOW ORDER BYTE POINTER
13CD A9 14      171            LDA /THEME2     ;SET MUSIC TABLE ADDRESS
13CF 85 09      172            STA PTR+1       ;TO HIGH ORDER BYTE POINTER
13D1 AE 17 15   173  MUSIC2    LDX MTEMP       ;INDEX X REGISTER
13D4 BD 63 14   174  MUSIC3    LDA THEME,X     ;LOAD FINE TUNE VALUE
13D7 8D A3 14   175            STA THEME2      ;STORE TO SOUND TABLE
13DA BD 83 14   176            LDA THEME1,X    ;LOAD COARSE TUNE VALUE
13DD 8D A4 14   177            STA THEME2+1    ;STORE TO SOUND TABLE
13E0 20 58 13   178            JSR START       ;PDODUCE SOUND
13E3 EE 17 15   179            INC MTEMP       ;INCREASE COUNTER
13E6 AD 17 15   180            LDA MTEMP       ;LOAD COUNTER VALUE
13E9 C9 20      181            CMP #$20        ;EQUAL TO 32?
13EB D0 02      182            BNE MUSIC4      ;NO, GET NEXT MUSIC NOTE
13ED A9 00      183            LDA #$00        ;YES, RESET COUNTER
13EF 60         184  MUSIC4    RTS             ;RETURN
```

大力水手

編者註：本遊戲同中加添有兩個程度的版本，一個是爲兒童而設，玩者在BOOT機後可選擇C（CHILDREN），按其他掣鍵則是ADULT版本，此版本中又由三個難度供選擇。

此外大家留意本遊戲是加有MOCKING BOARD 版本，有MOCKINGBOARD 的朋友請留意。

```
13F0            ;
13F0            ;
13F0            ; MUSIC AFTER POPYE ENERGISED
13F0            ;
13F0            ;
13F0 A9 00      MUSIC.1  LDA #$00
13F2 8D 18 15            STA MTEMP1
13F5 A9 A3      MUSIC.2  LDA #THEME2
13F7 85 08               STA PTR
13F9 A9 14               LDA /THEME2
13FB 85 09               STA PTR+1
13FD AE 18 15   MUSIC.3  LDX MTEMP1
1400 BD B3 14   MUSIC.4  LDA THEME3,X
1403 8D A3 14            STA THEME2
1406 BD D5 14            LDA THEME4,X
1409 8D A4 14            STA THEME2+1
140C 20 58 13            JSR START
140F EE 18 15            INC MTEMP1
1412 AD 18 15            LDA MTEMP1
1415 C9 22               CMP #$22
1417 D0 05               BNE MUSIC.5
1419 A9 00               LDA #$00
141B 8D 18 15            STA MTEMP1
141E 60         MUSIC.5  RTS
141F            ;
141F            ; MUSIC NOTE GENERATED AFTER
141F            ; POPYE GOT FALLEING OBJECTS
141F            ;
141F            ;
141F A9 F7      MUSIC.6  LDA #THEME5
1421 85 0A               STA PTR2
1423 A9 14               LDA /THEME5
1425 85 0B               STA PTR2+1
1427 A5 EF               LDA MTEMP2
1429 8D F7 14            STA THEME5
142C 06 EF               ASL MTEMP2
142E A5 EF               LDA MTEMP2
1430 8D 65 14            STA THEME+2
1433 20 74 13            JSR START2
1436 60                  RTS
1437            ;
1437            ;
1437            ; SOUND GENERATED WHEN POPYE
1437            ; HITS BIG MAN AFTER ENERGISED
1437            ;
1437            ;
1437 A9 07      GONG     LDA #GON
1439 85 0A               STA PTR2
143B A9 15               LDA /GON
143D 85 0B               STA PTR2+1
143F 20 74 13            JSR START2
1442 60                  RTS
1443            ;
1443            ;
1443            ;
1443 00 00 00   EXPLO    HEX 0000000000001027
1446 00 00 00
1449 10 27
144B 10 10 10            HEX 1010100030000000
144E 00 30 00
1451 00 00
1453            ;
1453            ;
1453 00 00 00   LTABLE   HEX 000000000000003E
1456 00 00 00
1459 00 3E
145B 0F 00 00            HEX 0F00000000000000
145E 00 00 00
1461 00 00
1463            ;
1463 8C 45 05   THEME    HEX 8C4505E8E8E8E8E8
1466 E8 E8 E8
1469 E8 E8
146B E8 E8 05            HEX E8E8054545454545
146E 45 45 45
1471 45 45
1473 45 45 8C            HEX 45458CDCDCDCDCDC
1476 DC DC DC
1479 DC DC
147B DC 45 DC            HEX DC45DC8C8C8C8C8C
147E 8C 8C 8C
1481 8C 8C
1483
1483 02 02 02   THEME1   HEX 0202020101010101
1486 01 01 01
1489 01 01
148B 01 01 02            HEX 0101020202020202
148E 02 02 02
1491 02 02
1493 02 02 02            HEX 0202020202020202
1496 02 02 02
1499 02 02
149B 02 02 02            HEX 0202020202020202
149E 02 02 02
14A1 02 02
14A3            ;
14A3 00 00 00   THEME2   HEX 000000000000003E
14A6 00 00 00
14A9 00 3E
14AB 10 00 00            HEX 100000000A000000
14AE 00 0A 00
14B1 00 00
14B3            ;
14B3            ;
14B3 C1 C1 C1   THEME3   HEX C1C1C1B7A3A3A3A3
14B6 B7 A3 A3
14B9 A3 A3
14BB 91 A3 7A            HEX 91A37A91A3A3A3A3
14BE 91 A3 A3
14C1 A3 A3
14C3 91 A3 7A            HEX 91A37A91A371C1C1
14C6 91 A3 71
14C9 C1 C1
14CB C1 C1 A3            HEX C1C1A3A391817A7A
14CE A3 91 81
14D1 7A 7A
14D3 7A 7A               HEX 7A7A
14D5            ;
14D5 00 00 00   THEME4   HEX 0000000000000000
14D8 00 00 00
14DB 00 00
14DD 00 00 00            HEX 0000000000000000
14E0 00 00 00
14E3 00 00
14E5 00 00 00            HEX 0000000000000000
14E8 00 00 00
14EB 00 00
14ED 00 00 00            HEX 0000000000000000
14F0 00 00 00
14F3 00 00
14F5 00 00               HEX 0000
14F7            ;
14F7 00 00 00   THEME5   HEX 000000000000003C
14FA 00 00 00
14FD 00 3C
14FF 10 10 00            HEX 1010000014000000
1502 00 14 00
1505 00 00
1507            ;
1507 00 00 00   GON      HEX 0000000000001007
150A 00 00 00
150D 10 07
150F 10 10 10            HEX 1010100030000000
1512 00 30 00
1515 00 00
1517            ;
1517            MTEMP    DFS 1
1518            MTEMP1   DFS 1
1519            ;
1519                     END

***** END OF ASSEMBLY
```

火樹銀花賀聖誕

黃博思



這個簡單可愛的程式能替你增添不少聖誕節氣氛。程式開始先用HPLOT繪出小屋及聖誕樹（白色是唯一令圖形不中斷的顏色），然後播出「平安夜」和用亂數在螢光幕繪出雪點。音樂會循環放送，直到你按［RESET］爲止。

音樂由LINE90－100的機械語言做出的新蘋果音樂指令產生；&TAEXPR 1，AEXPR 2。AEXPR 1是音調，AEXPR 2是拍子，兩者必需在0－255範圍內。蘋菓HARDWARE MANUAL P，132，TABLE 14指出＄3F5具有一跳越（JUMP）指令，以處理APPLESOFT中的"&"指令。產生音樂的程式存在＄300中，在＄3F5中存入JMP＄300。於是，每次& AEXPR 1，AEXPR 2在BASIC中執行，便播出音樂。由於過程極快，故在同時執行繪畫迴路時音樂效果仍具水準。音表請參閱表（1）。

音樂由LINE 160－170的DATA控制，要令音樂循環奏出，便需要一個改良的RESTORE指令。在MBASIC中，RESTORE可以指定從那一行開始，APPLESOFT的RESTORE實在把＄7D，＄7E的DATA POINTER重設在程式起點。POKE 125，97；POKE 126，11（LINE 140）把DATA POINTER重設到LINE 160，即音樂DATA的起點。注意在輸入程式時要完全依照列表LISTING，因爲程式的長度和安排會影响DATA在記憶體中的位址。如果眞的要改動程式，可加入LINE 115：PRINT PEEK（125），PEEK（126）：TEXT：STOP，察看＄7D，＄7E在讀入DATA前的數值，再代回LINE 140中。■

聖誕快樂…………！

表(1)

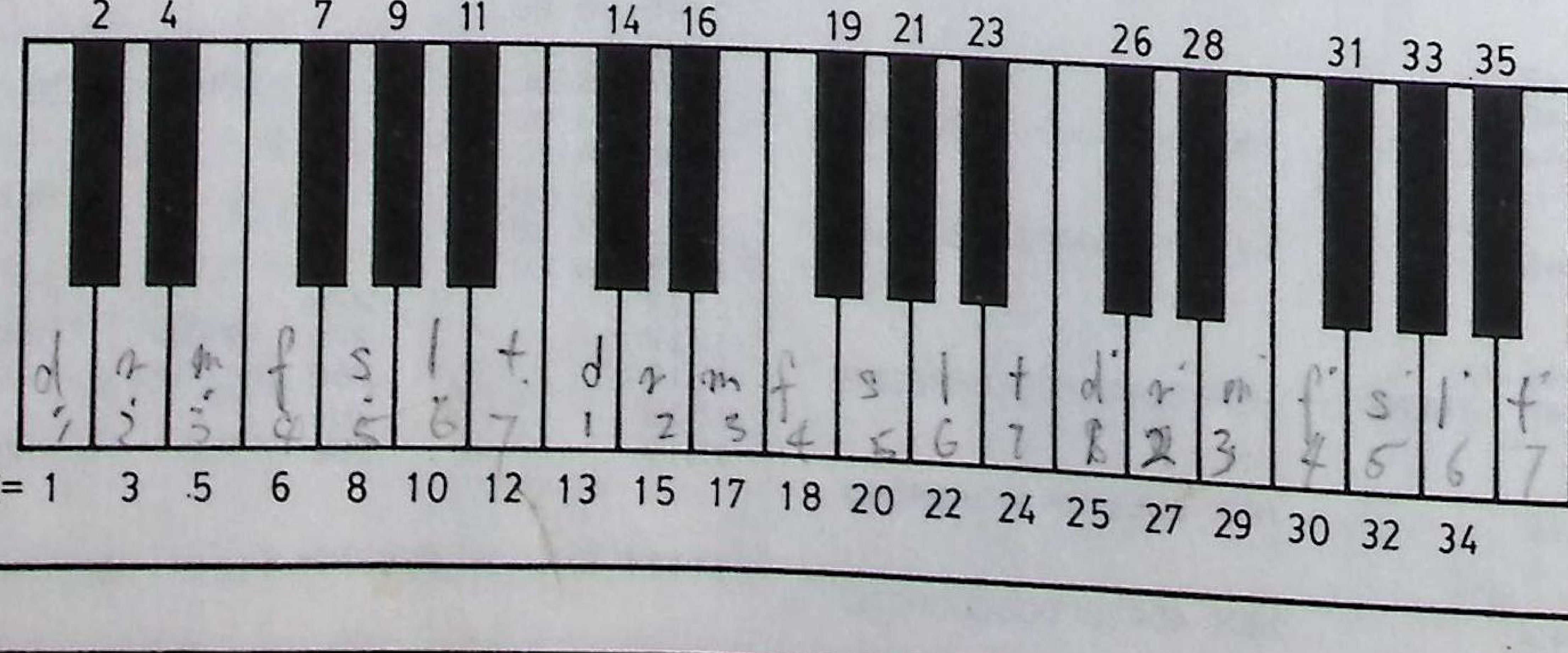
$d = DN$

$♩ = \frac{DN}{2}$

$♪ = \frac{DN}{4}$

$♬ = \frac{DN}{8}$

$0 < DN \leqslant 255$



火樹銀花賀聖誕

```
]LIST

10  REM  CHRISTMAS EVE:-MUSIC DIS
     K
20  REM     - WONG POK SZE 1984 DE
     C -
30  DIM P(45): TEXT : HOME : HTAB
     3: VTAB 22: PRINT "MERRY CHR
     ISTMAS AND HAPPY NEW YEAR": HTAB
     30: VTAB 24: PRINT "-P.S.WON
     G": HGR : HCOLOR= 3: ONERR  GOTO
     140
40  REM  DRAW BACKGROUND HOUSE
50  HPLOT 60,60 TO 100,20 TO 140,
     60: HPLOT 70,50 TO 70,130: HPLOT
     130,50 TO 130,130: HPLOT 20,
     130 TO 220,130
60  HPLOT 80,50 TO 90,50 TO 90,70
     TO 80,70 TO 80,50: HPLOT 85
     ,50 TO 85,70: HPLOT 80,60 TO
     90,60: HPLOT 95,130 TO 95,75
     TO 110,75 TO 110,130
70  HPLOT 40,130 TO 40,110 TO 25,
     110 TO 43,10 TO 61,110 TO 45
     ,110 TO 45,130
80  REM  MUSIC GENERATOR CODE
90  FOR R = 1 TO 45: READ P(R): NEXT
     : FOR I = 1 TO 66: READ P: POKE
     I + 767,P: NEXT : POKE 1013,
     76: POKE 1014,0: POKE 1015,3
     : DATA 192,181,171,161,152,1
     44,136,128,121,114,108,102,9
     6,91,86,81,76,72,68,64,60,57
     ,54,51,48,45,42,39,37,35,32,
     31,28,27,26,24,21,21,20,19,1
     8,17
100 DATA  16,15,14,201,84,208,15
     ,32,177,0,32,248,230,138,72,
     32,183,0,201,44,240,3,76,201
     ,222,32,177,0,32,248,230,104
     ,134,3,134,1,133,0,170,160,1
     ,132,2,173,48,192,136,208,4,
     198,1,240,7,202,208,246,166,
     0,208,239,165,3,133,1,198,2,
     208,241,96
110 REM  SNOW AND MUSIC
120 HPLOT  RND (1) * 279, RND (1
     ) * 159: READ NE,DN: & TP(NE
     ),DN / 2: & TP(NE),DN / 2
130 GOTO 120
140 POKE 125,97: POKE 126,11: RESUME
150 REM  SILENT NIGHT
160 DATA  8,150,10,25,8,50,5,150
     ,8,150,10,25,8,50,5,150,15,1
     00,15,50,12,250,13,100,13,50
     ,8,250
170 DATA  10,100,10,50,13,150,11
     ,25,10,50,8,250,10,25,8,50,5
     ,100,10,100,10,50,13,150,11,
     25,10,50,8,150,10,25,8,50,5,
     100,15,100,15,50,18,150,15,2
     5,12,50,13,250,17,100,13,150
     ,8,25,5,50,8,150,6,25,3,150,
     1,150,1,150
```

PRODOS 縱橫

PRODOS的PREFIX命令

普魯斯

APPLE公司在83年10月1日所推出的PRODOS系統裏，BASIC. SYSTEM 在處理PREFIX 命令時會出現一些錯誤，例如它在呼叫MLI（MACHINE LANGUAGE INTERFACE）時，緩衝器會存入一片空白。上期我曾發表了一些機械碼的進入點，現在我們利用這些資料來解決MLI這個問題，主要因爲MLI這個錯誤導致BASIC程式在OPEN一個TEXT FILE 時會出現INVALID PATHNAME SYNTAX（CODE＝40）。

要解決以上問題，在BASIC 程式呼叫機械語言程式之前，加入下列一行：

```
PRINT CHR$ (4) "PREFIX" : INPUT
PR$ : PRINT
CHR$ (4) "PREFIX" ; PR$
```

這樣做可以將PREFIX 設定而不會在MLI緩衝器上出現空白。閣下大可以將以上一行程式放在BASIC 程式的開首處。不過假如你要利用機械語言做啓動程式的話，我們需要加入一段相當於上列敍述句子的機械語言程式，這個程式由＄300開始，原理是呼叫MLI之後，利用MLI輸出的訊息來設定＄PREFIX，程式由＄341至＄380 就是這個作用。此時一PREFIX 已設定爲與VOLUME 相同。■

機械語言入門 吳博士

加減乘除運算原理(下)

ASL及ROL命令的功用

ASL命令是ACCUMULATOR SHIFT LEFT 的縮寫，它亦可以將記憶體的內容（八個BIT）向左移，不論是累加器或記憶體內容，當它們經左移之後，最右邊的一個BIT會以零來填補，以下的圖將各個BIT以1至8來代表以方便說明，當然在實際情形裏面這八個BIT 不是1便是0。

[ ] CARRY BIT

8	7	6	5	4	3	2	1

在ASL命令執行之前

[8] CARRY BIT

7	6	5	4	3	2	1	

在ASL命令執行之後

這個BIT被填入零。

從以上的圖可以看到，ASL命令會佔用CARRY BIT，因為原本的第8個BIT會被推上CARRY BIT之上。

ROL命令是ROTATE BITS TO THE LEFT 的縮寫，它可以在不同的位址模式之上使用，同樣ROL 命令會佔用CARRY BIT，和ASL 命令十分相像，不過最右邊的BIT不被填入零，而是填入CARRY BIT 的內容，請參考下圖：

[9] CARRY BIT

8	7	6	5	4	3	2	1

在ROL命令執行前

[8] CARRY BIT

7	6	5	4	3	2	1	9

在ROL命令執行之後

以上這兩個命令我們聯合來使用在兩個位址之上時便可將CARRY BIT 傳到第二個位址之上，例如以下的例子：

```
ASL   $393
ROL   $392
```

在執行之前

$392內容 | CARRY BIT | $393內容

F	E	D	C	B	A	9	8

[ ] CARRY BIT

7	6	5	4	3	2	1	0

在執行ASL $393之後

[7] CARRY BIT

F	E	D	C	B	A	8	9

6	5	4	3	2	1	0	

在執行ASL $393，ROL $392 之後

[F] CARRY BIT

E	D	C	B	A	9		7

6	5	4	3	2	1	0	

上面的執行過程我們可以歸納如下：

ASL $393將$393 之內容左移，而原本最左邊的內容被傳送到CARRY BIT 之內，而$393 最右邊的BIT會被零填入。

ROL $392將$392之內容左移，先把CARRY BIT上之數值填入$392 最右邊的BIT內，而同時最左邊的BIT傳入CARRY BIT中，這樣我們可以處理進位問，利用到乘法運算之中。

列表一就是利用以上的基礎編寫成的乘法程式，我們將要乘的數放在$393 及$394 之上，當然你不能夠相乘兩個兩位以上的數字。（即最多為兩位數字）計算結果會被存放到$390及$391之中。

列表(一)

```
300: LDA #$ 00
     STA  $390
     STA  $391
     STA  $392
     LDY #$ 00
     INY
     ROR  $394
     BCC  $326
     CLC
     LDA  $393
     ADC  $391
     STA  $391
     LDA  $392
     ADC  $390
     STA  $390
     ASL  $393
     ROL  $392
     CPY #$ 08
     BNE  $30D
     LNA  $390
     JSR  $FDDA
     LDA  $391
     JSR  $FDDA
     RTS
```

在上一期我們曾經討論過利用加法來作乘法運算及解釋過具體原理，現在乘法已告一段落，我們是否可以利用以上原理來處理除法呢？答案是可以的，因為除法事實上就是計算

加減乘除四則運算

被除數至多可被除數減幾次的運算，這個次數稱爲商數，而最後一次減法所剩下的數稱爲餘數。例如將1265除以13，用紙和筆來計算得到下列的結果：

```
       0097    ←—商數
 13 ) 1265
     -00
       12
     - 0
       126
      -117
         95
         91
          4    ←—餘數
```

以上的除法原理雖然很簡單，但是電腦就是這樣才能執行除法運算。和乘法一樣，6502 裏並沒有除法指令。在第十二期所介紹過的乘法是用一連串移位和加法運算而達成的。同理除法也可以用一連串減法運算做到。明確一點，就是除法可以用一連串所謂「嘗試減法」及以往我們討論過的比較指令來做到。在這些嘗試減法中，先將部份被除數與除數做比較（以被除數的最左邊開始），然後決定是否需要相減，現在我將剛才的例子作一說明。

$13_{10} = 1101$

$1265_{10} = 10011110001$

```
       00001100001
1101 ) 10011110001
      -0
       10
      - 0
       100
      -  0
       1001
      -   0
       10011
      - 1101
         1101
      -  1101
            01
      -      0
            010
      -       0
            0100
      -        0
            01000
      -         0
            010001
      -       1101
               100
```

圖 A

① 1 可以被13減幾次？答案是 0 次，所以在商放一個 0，然後取出下一個數字 2，使部份被除數變成12。

② 12可以被13減幾次？答案仍是 0 次，所以在商位放一個 0，然後取出下一個數字 6，使部份被除數變成126。

③ 126 能夠被13減幾次？答案是 9 次，所以在商放一個 9，然後將126減去117 （9 個13）。

④ 上述減法之餘數爲 9，所以取出最後一個數字 5，使部份被除數爲95。

⑤ 95可以被13減去幾次？答案是 7 次，所以商放一個 7，然後將95減去91（7 個13）。

⑥ 所有被除數的位數都被檢驗過後，整個除法則告完成，我們得到結果商數爲97（0097），餘數爲 4。

若果將以上除法改爲二進制計算，可以更爲簡單，因爲在二進位裏，商數字不是 1 就是 0，如果被除數可被除數減掉，則商的位數爲 1，否則爲 0，剛才的例子改爲二進制運算可以如上圖 A。

從以上可以看到二進位運算和十進位一樣，從後處理的商位數必須放在前一商位數的右邊，如果用程式來設計這種工作，可將原有部份的商向左移一位元，然後用一位新的商數值來塡補最右位元的空格，同時程式也必須使被除數向左移位，以形成和除數相比較的部份被除數。利用組合語言來執行二進位數除法的基本原理如下：

1.將商（起初爲零）左移一位，以空出下一商位數的位置。
2.將被除數左移一位，使最左一位元（LSB）移往進位BIT
3.將部份被除數向左旋轉，由進位BIT中取入新位元。
4.做除數和部份被除數之比較。
5.如果除數小於或等於部份被除數，則執行減法運算並在商位置內放入 1。
6.如果被除數內還剩下任何位元值，跳回步驟①。

以下的例子是利用位址＄11爲被除數而＄10爲除數的 8 位元不帶符號除法程式，商和餘數分別存於位址＄11和＄12之中。■

```
        LAD #0
        LDX #8
NXTBT   ASL $11
        ROL A
        CMP $10
        BCC CNTDN
        SBC $10
        INC $11
CNTDN   DEX
        BNE NXTBT
        STA $12
        BRK
```

物競天擇　適者生存

達爾文進化論模擬驗証

四方帽

相信大家同學都讀過有關達爾文的進化論。 物競天擇，適者生存 。達爾文認爲人類是進化而來的。科學家亦進行了許多實驗來試圖證明這理論，但目前仍有許多謎未能解開以證實這理論的眞確性。進化論最主要的一項論據是或然率，單細胞在幾千百萬份之一的機會下始能進化爲複細胞，複細胞又在同等的情況下再進化。

爲了讓大家對進化論的概念有更深的認識，我編寫了一個〔物競天擇〕的模擬進化論程式。當然這個模擬程式是十分原始，原因在實際的情況是過於複雜，根本不能模擬。

模擬程式假設了有100 種生物（由1至100），每種生物有五種遺傳因子，分別由１至５代表。而每種生物的遺傳因子所代表的數目總和是該生物能夠生存下去的因素，並將這數值和１至30之中隨機出現的一個數值比較，假如這個隨機數值是大過生存因素數值，則該生物便會“死亡”，取而代之的是另一種生物。新的生物本身的特性是由任何其他兩種的生物而決定，而他的遺傳因子是由其“父母”的相對遺傳因子所決定。於是一種新的“後代”便產生了。假如隨機數值是細於遺傳因素總和，則該生物便能繼續生存下去，沒有任何事發生。由此而得知，有較細數值的遺傳因素其被殺的機會越大。程式生存因素的平均值會每週顯示一次。假如有生物進化，則此生存因素的平均值會漸次接近最高值。經過約二百星期後，程式會將平均生存因素值作成曲線圖，供參考比較。或者，各同學們可以在程式進行期間按下〔S〕鍵停止。

在某個程度上來說，此程式可以將達爾文的進化論精髓：物競天擇，適者生存實踐出來。各位同學不妨試將這程式鍵入，觀看其執行結果。老師們亦可利用這個程式來對學生驗證這套理論。■

```
 5  REM  :::::::::::::::::::::::::
 10  REM
 15  REM     A DEMONSTRATION
 20  REM     OF EVOLUTION DUE
 25  REM   TO NATURAL SELECTION
 30  REM
 35  REM  :::::::::::::::::::::::
     :::
 100  TEXT : HOME
 105  DIM GENES%(100,5),WEEKDTA(20
     0)
 110 NUMBER = 100
 115  REM  :
 120  REM  : CREATE 100 ANIMALS
 125  REM  :
 130  FOR ANIMAL = 1 TO NUMBER
 135 GENES%(ANIMAL,0) = 1
 140  FOR GENE = 1 TO 5
 145  VTAB 8: PRINT "    CREATING
      ANIMAL NUMBER ";ANIMAL
 150 GENES%(ANIMAL,GENE) =  INT ( RND
     (1) * 5) + 1
 155  NEXT GENE,ANIMAL
 160  HOME : VTAB 24: PRINT "*****
     ******************************
     *******": VTAB 12: PRINT "**
     ******** PRESS 'S' TO STOP *
     ***********": VTAB 1: PRINT "
     ******************************
     *************"
 165  VTAB 3: PRINT "WEEK ";: HTAB
     20: PRINT "POPULATION: "
 170  VTAB 5: PRINT "ANIMAL "
 175  VTAB 7: PRINT "ANIMAL SURVIV
     ABILITY FACTOR: "
 180  VTAB 8: PRINT "RANDOM SURVIV
     ABILITY FACTOR: "
 185  VTAB 14: PRINT "DATA FOR WEE
     K #"
 190  VTAB 17: PRINT "AVERAGE SURV
     IVABILITY FACTOR:"
 195  REM  :
 200 WEEK = 1
 205  REM  :
 210  REM  :NOW, CALCULATE AND
 215  REM  :OUTPUT THE STATISTICS
 220  REM  :FOR THIS ANIMAL
 225  REM  :POPULATION
 230  REM  :
 235 TSUM = 0:TNUM = 0
 240  FOR ANIMAL = 1 TO 100
 245  IF GENES%(ANIMAL,0) = 0 THEN
     280
 250 SUM = 0
 255  FOR GENE = 1 TO 5
 260 SUM = SUM + GENES%(ANIMAL,GEN
     E)
 265  NEXT GENE
 270 TSUM = TSUM + SUM
 275 TNUM = TNUM + 1
 280  NEXT ANIMAL
 285  REM  :
 290  REM  : OUTPUT THE RESULTS
 300  VTAB 14: HTAB 15: PRINT WEEK
 305 WKVL =  INT (TSUM * 100 / TNU
     M) / 100
 310  VTAB 17: HTAB 30: PRINT  INT
     ((WKVL - OLDWKVL) * 100) / 1
     00;: CALL  - 868
```

```
320 WEEKDTA(WEEK) = WKVL
325 OLDWKVL = WKVL
330  REM  :
335  REM  : GO ONTO THE NEXT WEEK

340  REM  :
345  FOR GENE = 1 TO 5
350  FOR ANIMAL = 1 TO 100
355  IF GENES%(ANIMAL,GENE) < >
     GENES%(1,GENE) THEN 375
360  NEXT ANIMAL
365  NEXT GENE
370  GOTO 650
375 GENE = 6:ANIMAL = 1000: NEXT
     ANIMAL,GENE
380  IF  PEEK ( - 16384) =  ASC (
     "S") + 128 OR WEEK = 200 THEN
     650
385  REM  :
390  REM  :AT LEAST ONE ANIMAL IS

395  REM  :DIFFERENT FROM THE RES
     T
400  REM  :SO GO ON TO THE NEXT
405  REM  :WEEK.
410  REM  :
415  REM  : NOW, LOOP THROUGH
420  REM  :POPULATION UNTIL
425  REM  :ALL OF THE ANIMALS
430  REM  : ARE EXACTLY THE SAME
435  REM  :
440  REM  :    IN EACH LOOP,
445  REM  :REMOVE ANY ANIMAL
450  REM  :IN WHICH A RANDOM
455  REM  :NUMBER FROM 1 TO 30
460  REM  :IS LARGER THAN THE
465  REM  :SUM OF THE VALUES OF
470  REM  :THE ANIMALS "GENES".
475  REM  :
480  VTAB 3: HTAB 6: PRINT WEEK;:
      HTAB 32: PRINT NUMBER; SPC(
     3);: PRINT
485  FOR ANIMAL = 1 TO 100
490  IF GENES%(ANIMAL,0) = 0 THEN
     640
495  VTAB 5: HTAB 12: PRINT ANIMA
     L; SPC( 3): PRINT
500 RANM =  INT ( RND (1) * 30) +
     1
505 SUM = 0
510  FOR TEMP = 1 TO 5
515 SUM = SUM + GENES%(ANIMAL,TEM
     P)
520  NEXT TEMP
525  VTAB 7: HTAB 30: PRINT SUM; SPC(
     3): PRINT : VTAB 8: HTAB 30:
      PRINT RANM; SPC( 3): PRINT

530  VTAB 15: HTAB 1: CALL  - 868

535  IF SUM > = RANM THEN 640
540  REM  :
545  REM  :THE CURRENT ANIMAL HAS
550  REM  :DIED, SO PUT A NEW ANI
     MAL
555  REM  :INTO THE DEAD ANIMAL'S
      PLACE
560  REM  :
565  REM  :THIS ANIMAL'S GENES WI
     LL
570  REM  :BE CREATED BY COMBING
575  REM  :THE GENES OF 2 OTHER A
     LIVE
580  REM  :ANIMALS, RANDOMLY USIN
     G
585  REM  :ONE OF THE TWO GENES I
     N
590  REM  :EACH LOCATION TO GET T
     HE
595  REM  :GENE FOR THE NEW ANIMA
     L.
600  REM  :
605 FTHER =  INT ( RND (1) * 100)
      + 1: IF GENES%(FTHER,0) = 0
      THEN 605
610 MOTHER =  INT ( RND (1) * 100
     ) + 1: IF GENES%(MOTHER,0) =
     0 OR MOTHER = FTHER THEN 605

615  FOR GENE = 1 TO 5
620  IF  RND (1) > .5 THEN 630
625 GENES%(ANIMAL,GENE) = GENES%(
     FTHER,GENE): GOTO 635
630 GENES%(ANIMAL,GENE) = GENES%(
     MOTHER,GENE)
635  NEXT GENE
640  NEXT ANIMAL
645 WEEK = WEEK + 1: GOTO 205
650  REM  :
655  REM  :ALL ANIMALS ARE THE SA
     ME,
660  REM  :SO END THE PROGRAM BY
665  REM  :PRINTING A GRAPH OF TH
     E
670  REM  :WEEKLY AVERAGES.
675  REM  :
680  HOME
685  PRINT "WEEK";: HTAB 6: PRINT
     "WEEKLY AVERAGE OF SURVIVAL
     TRAIT"
690  FOR LOOP = 1 TO WEEK
695  PRINT LOOP;: HTAB 5
700 STARS = WEEK(LOOP) * 35 / 25
705  FOR TEMP = 1 TO STARS
710  PRINT "*";
715  NEXT TEMP
720  PRINT
725  NEXT LOOP
730  PRINT : INPUT "SHOW GRAPH AG
     AIN?";ANSWER$:ANSWER$ = ANSW
     ER$ + " "
735  IF  LEFT$ (ANSWER$,1) = "Y" OR
      LEFT$ (ANSWER$,1) = "y" THEN
     650
740  VTAB 24: PRINT : PRINT : PRINT
     : END
```

低解像圖輔助繪劃器

周美新

編寫高解像圖形程式並非一項一學即會的事，需要浸淫長時間方成，由於要計算記憶位址，所以需要有良好的數學基礎。對於兒童來說，更是困難重重。不過，假如由低解像圖形學起，則可大大提高他們的興趣。繪畫低解像圖形並不需要太多的技巧，只要弄清X，Y座標代表的螢幕位址關係，則很容易掌握繪圖的技巧。

不過，話由說回來，要讓6至12歲的兒童使用VLIN及HLIN 敍述句來繪畫低解像圖，如果不是有長時間接觸電腦亦怕不易做到。

筆者於是設計了這個低解像圖輔助繪劃器（LO—RES DRAWING）給這階段年紀的電腦兒童，利用這程式可以一面繪畫，一面看繪出來的結果。如此，不但可讓兒童加強繪圖的興趣，而且更能培養日後對電腦程式編寫的技能。

透過本程式，兒童可輕易在低解像頁裏設定水平及垂直圖線並指定其顏色，所用的指令跟BASIC 中所用的一樣，但過程就簡化得多，而且繪出的圖可以儲存在磁碟上，留待日後取用或以印字機編印出來（編印時必須要有GRAPPLER咭或其他有傾印低解像圖功能的印字機控制咭）。

程式的控制鍵相當容易記，包括有：

C＝COLOR	指定色彩
V＝VLIN	畫垂直線
H＝HLIN	畫水平線
D＝DELETE	取消最後繪畫的一條線
R＝RE—DRAW	清理螢幕，然後再將所劃的線劃出
S＝SAVE	將圖像存起在磁碟
L＝LOAD	從磁碟載入低解像圖檔案
P＝PRINT	將圖像印出
Q＝QUIT	離開程式

？＝顯示各個控制指令

當程式進行時，如要觀看磁碟上的目錄，可隨時打入CATALOG，觀看各檔案。

各位老師或家長，希望你們能好好的利用這個程式來誘導兒童們來加強他們對繪畫圖像的興趣。為了讓你們可將繪畫的指令及數據紀錄下來，筆者另外再寫了一個短程式(見列表DRAWING FILE）來將圖像檔案列表。■

```
 10  REM  ::::::::::::::::
 15  REM   LO-RES DRAWING
 20  REM         BY
 25  REM   CHOW MEI SHUN
 30  REM   COPYRIGHT (C)
 35  REM       1984
 40  REM    COMPUHOUSE
 45  REM  ::::::::::::::::
100  GR : VTAB 21: HTAB 1: CALL  -
     958: DIM D%(500,4):N = 0:CL =
     15:D$ =  CHR$ (4)
110  DIM B%(15,3): FOR Z = 0 TO 1
     5: FOR K = 0 TO 3: READ B%(Z
     ,K): NEXT K,Z: REM  DOT PATT
     ERN FOR HARD COPY PRINT OUT
120  DATA  0,0,0,0,18,0,18,0,34,8
     ,17,4,21,0,21,0,18,36,18,9,1
     8,18,18,18,18,37,18,41,21,42
     ,21,42,21,21,21,21,63,0,63,0
     ,63,16,63,4,63,10,63,20,27,5
     4,27,54,27,27,27,27,63,21,63
     ,42,63,63,63,63
130  VTAB 21: HTAB 1: CALL  - 958
     : PRINT "           LO-RES GRAP
     HICS PLOTTER": PRINT "->";: POKE
     49168,0
140  GET KB$: ON KB$ <  > "Q" GOTO
     150: POKE 49168,0: INPUT "QU
     IT (CONFIRM - Y/N)? ";YN$: ON
      LEFT$ (YN$,1) <  > "Y" GOTO
     130: TEXT : HOME : END
150  ON KB$ <  > "V" AND KB$ <  >
     "H" GOTO 210: PRINT KB$;"LIN
     ";:D%(N,0) = (KB$ = "V")
160  VTAB 22: HTAB 8: INPUT "FROM
     : ";KB$: ON  LEN (KB$) = 0 GOTO
     130:L1 =  VAL (KB$): IF L1 <
      >  INT (L1) OR L1 < 0 OR L1
      > 39 THEN  PRINT  CHR$ (4):
      GOTO 160
170  VTAB 22: HTAB 18: INPUT "TO:
      ";KB$: ON  LEN (KB$) = 0 GOTO
     160:L2 =  VAL (KB$): IF L2 <
      >  INT (L2) OR L2 < 0 OR L2
      > 39 THEN  PRINT  CHR$ (4):
      GOTO 170
180  VTAB 22: HTAB 30: INPUT "AT:
      ";KB$: ON  LEN (KB$) = 0 GOTO
     170:L0 =  VAL (KB$): IF L0 <
      >  INT (L0) OR L0 < 0 OR L0
      > 31 THEN  PRINT  CHR$ (4):
      GOTO 180
190 D%(N,1) = CL:D%(N,2) = L0:D%(
     N,3) = L1:D%(N,4) = L2:L = N
     : GOSUB 460:N = N + 1: ON N <
     300 GOTO 130
200  VTAB 23: HTAB 1: PRINT "STOR
     AGE SPACE IS FULL":: END
210  ON KB$ <  > "C" GOTO 230: POKE
     49168,0: INPUT "COLOR= ";YN$
     : ON  LEN (YN$) = 0 GOTO 130
     :L0 =  VAL (YN$): IF L0 <  >
      INT (L0) OR L0 < 0 OR L0 >
     15 THEN  VTAB 22: HTAB 3: CALL
      - 958: GOTO 210
220 CLR = L0: GOTO 130
230  ON KB$ <  > "D" GOTO 250: PRINT
     "DELETE LAST LINE":N = N - 1
     : IF N < 0 THEN N = 0: GOTO
     130
240  GOTO 260
```

```
250  ON KB$ <  > "R" GOTO 270: PRINT
     "REDRAW LINES SO FAR.": IF N
      < 1 THEN  GR : GOTO 130
260  GR : FOR L = 0 TO N - 1: GOSUB
     460: NEXT L: GOTO 130
270  ON KB$ <  > "S" GOTO 320: POKE
     49168,0: INPUT "SAVE IN FILE
     :";FL$: ON  LEN (FL$) = 0 GOTO
     130: ONERR  GOTO 300
280  IF FL$ = "CATALOG" THEN  GOSUB
     440: GOTO 260
290  PRINT D$"VERIFY "FL$: POKE 2
     16,0: VTAB 23: HTAB 1: PRINT
     "THAT NAME HAS BEEN USED. TR
     Y AGAIN.": VTAB 22: HTAB 3: CALL
      - 868: GOTO 270
300  POKE 216,0: IF  PEEK (222) <
      > 6 THEN  VTAB 23: HTAB 1: PRINT
     "UNABLE TO ACCES DISK. FIX P
     ROGRAM.": VTAB 22: HTAB 3: CALL
      - 868: GOTO 270
310  PRINT D$"OPEN "FL$: PRINT D$
     "WRITE "FL$: PRINT N: FOR L =
     0 TO N - 1: FOR J = 0 TO 4: PRINT
     D%(L,J): NEXT J,L: PRINT D$"
     CLOSE "FL$: GOTO 130
320  ON KB$ <  > "L" GOTO 380: POKE
     49168,0: INPUT "LOAD FROM FI
     LE: ";FL$: ON  LEN (FL$) = 0
      GOTO 130: ONERR  GOTO 360
330  IF FL$ = "CATALOG" THEN  GOSUB
     440: GOTO 260
340  PRINT D$"VERIFY "FL$: POKE 2
     16,0: PRINT D$"OPEN "FL$: PRINT
     D$"READ "FL$: INPUT K: IF K +
     N > 500 THEN  PRINT D$"CLOSE
     ": VTAB 23: HTAB 1: PRINT "
     NOT ENOUGH ROOM TO LOAD.": FOR
     J = 1 TO 2000: NEXT J: GOTO
     130
350  FOR L = N TO N + K - 1: FOR
     J = 0 TO 4: INPUT D%(L,J): NEXT
     J,L: PRINT D$"CLOSE "FL$:N =
     N + K: GOTO 260
360  POKE 216,0: VTAB 23: HTAB 1:
      IF  PEEK (222) <  > 6 THEN
      PRINT "UNABLE TO ACESS DISK
     . FIX PROBLEM.": VTAB 22: HTAB
     3: GOTO 320
370  VTAB 23: HTAB 1: PRINT "THAT
     FILE ISN'T ON THE DISK. TRY
     AGAIN.": VTAB 22: HTAB 3: GOTO
     320
380  ON KB$ <  > "P" GOTO 420: POKE
     49168,0: INPUT "PRINT - IS P
     RINTER READY (Y/N) ?";YN$: IF
     LEFT$ (YN$,1) <  > "Y" THEN
     130
390  PRINT  CHR$ (4)"PR#1": PRINT
     : PRINT  CHR$ (27)"T12": FOR
     Y = 0 TO 39: PRINT  CHR$ (27
     )"S0320";: FOR X = 0 TO 39:Z
     =  SCRN( X,Y): FOR J = 1 TO
     2: FOR K = 0 TO 3
400  ON  PEEK (49601) > 127 GOTO
     400: POKE 49296,B%(Z,K): NEXT
     K,J: NEXT X: PRINT : NEXT Y:
     PRINT  CHR$ (27)"B": PRINT
     : PRINT  CHR$ (4)"PR#0"
410  GOTO 130
420  ON KB$ <  > "?" GOTO 130: PRINT
     "COMMANDS: V=VLIN H=HLIN C=C
     OLOR Q=QUIT D=DELETE R=REDRA
     W S=SAVE L=LOAD P=PRINT  ";:
      INVERSE : PRINT " PRESS SPA
     CE BAR TO CONTINUE ";: NORMAL
     : POKE 49168,0
430  GET KB$: ON KB$ <  > " " GOTO
     430: GOTO 130
440  TEXT : HOME : PRINT : PRINT
      CHR$ (4)"CATALOG": PRINT : PRINT
     : INVERSE : PRINT " PRESS SP
     ACE BAR TO CONTINUE ";: NORMAL
     : POKE 49168,0
450  GET OK$: ON OK$ <  > " " GOTO
     450: HOME : RETURN
460  COLOR= D%(L,1): IF D%(L,0) =
     1 THEN  VLIN D%(L,3),D%(L,4)
      AT D%(L,2): RETURN
470  HLIN D%(L,3),D%(L,4) AT D%(L
     ,2): RETURN
```

```
10   REM ::::::::::::::::
15   REM    DRAWING FILE
20   REM         BY
25   REM   CHOW MEI SHUN
30   REM   COPYRIGHT (C)
35   REM        1984
40   REM    COMPUHOUSE
45   REM ::::::::::::::::
1000 D$ =  CHR$ (4): TEXT : HOME
     : INPUT "FILENAME: ";FL$: IF
     LEN (FL$) = 0 THEN  END
1010 IF FL$ = "CATALOG" THEN  PRINT
     D$$"CATALOG": PRINT : PRINT
     : PRINT : INPUT "PRESS RETUR
     N TO CONTINUE:";KB$: RUN
1020 ONERR  GOTO 1040
1030 PRINT D$"OPEN "FL$: PRINT D
     $"READ "FL$: INPUT N: DIM D%
     (N,4): FOR J = 1 TO N: FOR X
     = 0 TO 4: INPUT D%(J,K): NEXT
     K,J: PRINT D$"CLOSE": GOTO 1
     050
1040 POKE 216,0: PRINT "UNABLE T
     O LOCATE OR ACCESS THE FILE.
     ": FOR J = 1 TO 3000: NEXT J
     : RUN
1050 POKE 216,0: PRINT : PRINT "
     HARDCOPY (Y/N)?"";: POKE 481
     69,0
1060 GET KB$: ON KB$ <  > "Y" AND
     KB$ <  > "N" GOTO 1060: PRINT
     KB$: IF KB$ = "Y" THEN  PRINT
     D$"PR#1"
1070 PRINT "THE DRAWING CONTAINS
     "N" LINES.": PRINT "DIRCLR
       FROM        TO        AT": FOR
     J = 1 TO N: PRINT  AT ": FOR
     J = 1 TO N: PRINT MID$ ("HV
     ",D%(J,0) + 1,1)""D%(J,1);:
     HTAB 19:PRINT D%(J,3);:HTAB
     28:PRITN D%(J,4);: HTAB 37:P
     RINT D%
1080 PRINT  CHR$ (4)"PR#0": END
```

為蘋果加上編輯功能

程式編輯器

張俊傑

程式編輯器是設計給予APPLE用家在設計，編輯，除錯時得到事半功倍之效果，大家都知道APPLE是沒有編輯功能的，因此我設計了這個PROGRAM EDITOR，相信在發展程式之時會收到很好的效果。

在以下的篇幅裏我將會討論程式編輯器的操作，其中最重要的是程式編輯器本身如何改良程式，及將ESC鍵的功能擴充。你只要按下兩個鍵就可以得到你一般要按入很多鍵才得到的資料，控制程式列表功能，接受小階輸入命令，及以程式來控制CURSOR的位置。

程式的工作環境

程式編輯器是設計在64K的系統上使用，它配合著在第十期發表的「將DOS移上RAM咭」一起工作，因此要執行PROGRAM EDITOR時必須先BRUN RAM DOS使標準的DOS系統移上RAM咭，然後再執行本程式編輯器（BRUN PROGRAM EDITOR），這個系統此時會佔用$B900開始$600個BYTE長，同時在啓動它之後會自動將HIMEM設定為$B900使不被BASIC執行後破壞，另一方面亦將RESET的指標更改，使無論何時按下RESET鍵都不致中斷本系統。這是十分好的安排，因為一般的DOS佔用$9600開始及以上的記憶體，使工作空間降低，因此有如此改善的必要。

當啓動本系統之後，你即可以對程式加以編輯，當你按下CTRC—E之後，螢幕上便會出現EDIT字樣，這時你按入要編輯的行號並按〔RET〕便可以進入編輯器之內，CURSOR會停在該行的第一個敍述句之上等待你進一步的操作，若你修改了其中的內容甚或沒有修改而按下〔RET〕，程式編輯器亦會將整個行號的內容抄到記憶體去，省却移動"→"鍵的麻煩。

PROGRAM EDITOR會記著最後一次EDIT（編輯）過的行號，因此若你希望EDIT同一行的話，只需按CTRL—E及一個句點"。"即可，這樣該行便會重現給你編輯。

注意事項

以下六點是以CTRL—E進入編輯模式內需要注意的事項：

1. CTRL—E必須是一行之中第一個按入的鍵。
2. APPLESOFT的第0行是不能使用本程式編輯器處理之。
3. 在APPLESOFT程式執行的過程之中，若有INPUT敍述句進行，要求輸入而給出CURSOR予用戶以下，CTRL—E亦會暫時失去作用，同樣在監督程式生效時，CTRL—E亦會失效。
4. 按下CTRL—E出現EDIT字樣，如在鍵入行號前按入"←"鍵到EDIT上去會導致CTRL—R（RESTART）功能（較後討論）喪失。

現在各位可以進入編輯模式了，當以CTRL—E進入後，尚有以下的命令可以使用：

(1) CTRL—I （INSERT），此命令可以將字符加入正在編輯的行裏，由CURSOR所在的位置開始插入，完成後按〔RET〕即可。

(2) CTRL—D（DELETE），此命令可以將正在編輯的行號裏，由CURSOR所在位置開始，每按一次CTRL—D便將一個字刪除。

(3) CTRL—F （FIND），按入CTRL—F後緊接一個字符可以快速移動CURSOR到按入的字符上作其他編輯工作。

(4) CTRL—O（OVERRIDE），這命令與CTRL—I極為相似，所不同的是用來插入一些CONTROL字符在程式行號之上。在使用時先按下CTRL—I再按CTRL—O緊接要插入的CTRL字元，若果還有的話，重覆這個步驟。

(5) CTRL—R（RESTART），這命令在正當編輯的行號中按入時，會不顧你改動過甚麽，將原來的行號再次出示在螢幕上予你編輯。

(6) CTRL—P（PACK），這個命令使你的程式行號中所有空隔都刪除，緊迫在一起，這命令最適宜用在很長的敍述句上編輯之用，例如我們的ONE—LINE—PROGRAM。

(7) CTRL—C（CONVERT），這個命令還是用來快速地將大階字元轉為小階之用的，或小階轉為大階之用，你可以預先設定顯示模式（以CTRL—A設定大階，CTRL—S設定為小階），之後以"→"鍵移過所需轉換的字元。（CTRL—S 設定小階後〔RET〕，CTRL—E〔NO〕〔RET〕CTRL—C，〔→〕〔RET〕同時按下。）

(8) CTRL—B（BEGIN），這個命令可以立即將CURSOR所在位置移到一行上最左位置，適宜配合CTRL—F命令或更改行號之用。

(9) CTRL—N（END），使CURSOR立即移到行號之末的位置。

(10) CTRL—Q（QUIT）這命令將CURSOR所在位置開始的一切刪除，將刪除後的資料輸入記憶並離開EDITOR。

(11) CTRL—M 相當於〔RET〕鍵。

(12) CTRL—X（CANCEL），取消正在編輯的行號，離開EDITOR，但記憶內的行號則不受影響。

(13) CTRL—A 設定大階字元輸入。

(14) CTRL—S 設定小階字元輸入。

程式 LIST 控制

由於EDITOR改良了鍵盤緩衝器，因此要暫停LIST功能可以按下ESC鍵，或以往的CTRL—S，而再按入任何鍵會使LIST 功能繼續。另一個新加入的功能是CTRL—F（FLUSH），它可以使快速移到一些行號去，而中斷在螢幕上的輸出，這對LIST一些很長的程式時會很有幫助。

編輯器與MACRO鍵盤

這個PROGRAM EDITOR的另一個卓越功能是MACRO KEYBOARD 效能，在一般操作時，只須按下ESC 鍵及另一個鍵緊接之後會使MACRO生效，這些鍵的功能如表㈠所列，希望各位讀者使用時能夠得心應手！■

```
REM   ::::::::::::::::::
REM    PROGRAM EDITOR
REM          BY
REM    CHEUNG CHUN KIT
REM    COPYRIGHT RIGHT
REM      (C) 1984 BY
REM     COMPUTING AGE
REM   ::::::::::::::::::
B8E3- 4C EF B8 00 B9
B8E8- 1C 4C 0C B9 4C 00 B9 A2
B8F0- 09 BD E5 B8 9D F1 03 CA
B8F8- D0 F7 86 73 A2 B9 86 74
B900- 4E AF BD 4E AE BD 20 0C
B908- B9 4C AE BF 20 12 B9 4C
B910- CC BF A9 3E A0 BD 20 39
B918- BD 4E B1 BD A9 23 A0 B9
B920- 4C 34 BD 4E B1 BD 20 E2
B928- BC C9 9B D0 68 20 DE BC
B930- AD 00 C0 09 80 85 3F A9
B938- B2 85 08 A9 BD 85 09 A0
B940- 00 B1 08 F0 10 C5 3F F0
B948- 16 20 58 B9 B1 08 30 F9
B950- 20 58 B9 D0 EC 4C 0C FD
B958- E6 08 D0 02 E6 09 60 20
B960- 58 B9 A9 6C A0 B9 20 34
B968- BD 4C 6E B9 91 28 4E AF
B970- BD A0 00 B1 08 10 11 20
B978- 58 B9 A4 24 C9 A7 D0 05
B980- 8D B1 BD F0 EC 4C 06 BD
B988- 48 20 19 B9 68 C9 7F F0
B990- 9C 09 80 D0 E5 C9 85 F0
B998- 28 C9 8D F0 04 C9 98 D0
B9A0- 03 8E B0 BD C9 97 D0 0A
B9A8- 46 3E E0 00 D0 08 24 33
B9B0- 50 01 60 AE B0 BD 86 45
B9B8- 20 8E FD 20 3C BA 4C 8F
B9C0- BA E0 00 D0 ED A4 33 C0
B9C8- DD F0 04 C0 BE D0 E3 A2
B9D0- 04 BD A3 BD 9D 00 02 BD
B9D8- 9E BD 20 ED FD CA 10 F1
```

編者按：假如各位是電腦時代的長期讀者，必會知道我們在第十期刊登有〔將DOS 移上RAM 咭〕，MARCO鍵盤，第六期的APPLESOFT程式偵錯系統；第十二期的比RAM DISK 更好用的RAM咭控制中心。再加上今期的程式編輯器，對於各位在發展程式或設計鍵入程式時，構成了一個非常強效的實用系統。

我們特別將這幾個程式收集在一張磁碟（編號CA00D）上方便有需要的讀者使用。各位可前來電腦時代讀者服務部購買，每張45元。或者用郵寄的方式購買，請用劃線支票，抬頭寫〔COMPUTING AGE PUBLISHER〕寄來九龍郵政信箱CPO BOX 71193，電腦時代出版社收。信封面請註明〔郵購〕。郵購CA00D免付郵費，如要掛號請加付4元。外埠讀者空郵訂購請加付20％。■

表①

```
ESC 1   CATALOG1
    2   CATALOG2
    3   LOAD
    4   SAVE
    5   BLOAD
    6   BSAVE
    7   LOCK
    8   UNLOCK
    9   DELETE
    10  VERIFY
    :   CALL-151
    C   CHR$( )
    H   POKE-16302,
        0:POKE -162
        97,0:POKE-1
        6304,0:POKE
        (-16301+P).
        0
```

使用時先鍵入所要觀看的高解像頁數入P內，例如P＝1：ESC H 即可觀看第一頁。

```
    L   LIST
    N   NOTRACE
    P   POKE
    Q   PRINT PEEK
        A)+(PEEK(A+
        1)*256
    R   RETURN
    T   TRACE
    W   BFFC.BFFF
```

```
B9E0- A2 05 20 75 FD E0 06 90
B9E8- 1A AD 05 02 C9 AE D0 0F
B9F0- A2 FF E8 BD A3 BD 9D 00
B9F8- 02 C9 8D D0 F5 F0 0E E0
BA00- 0B 90 0A 20 DD FB A2 00
BA08- 86 45 A9 8D 60 A2 05 BD
BA10- 00 02 9D A3 BD C9 8D F0
BA18- 0F C9 A0 F0 08 C9 B0 90
BA20- E2 C9 BA B0 DE E8 D0 E7
BA28- 85 3E A9 48 A0 BA 20 39
BA30- BD A9 75 20 34 BD 46 3D
BA38- A9 01 85 45 A9 EE 24 33
BA40- 70 02 A9 7F 85 44 D0 C2
BA48- C9 8D F0 C0 48 8D 00 02
BA50- A9 62 85 36 24 33 50 08
BA58- A9 ED 85 B8 A9 F9 85 B9
BA60- 68 60 86 43 A6 45 9D 00
BA68- 02 E6 45 E4 44 90 03 20
BA70- 8D BC A6 43 60 C6 45 D0
BA78- 03 4C 32 BB 24 3D 10 03
BA80- 20 8D BC A6 45 A9 8D CA
BA88- DD FF 01 F0 FA 86 45 A9
BA90- 0C A0 BB 20 34 BD A9 1E
BA98- A0 BD 20 39 BD A2 00 20
BAA0- BE BA E4 45 B0 15 BD 00
BAA8- 02 C9 A0 F0 08 C9 B0 90
BAB0- 0A C9 BA B0 06 20 1E BD
BAB8- E8 D0 E7 4C 0C FD 86 43
BAC0- 20 E2 BA E4 45 B0 09 BD
BAC8- 00 02 20 EB BA E8 D0 F3
BAD0- A9 A0 20 EB BA A4 41 84
BAD8- 24 A5 42 85 25 A6 43 4C
BAE0- C1 FB A5 24 85 41 A5 25
BAE8- 85 42 60 A4 24 C8 C4 21
BAF0- 90 09 A4 25 C8 C4 23 90
BAF8- 02 C6 42 4C 1E BD E4 45
BB00- F0 09 BD 00 02 20 EB BA
BB08- E8 D0 F3 60 20 E2 BC 86
BB10- 43 E4 45 90 02 86 45 A4
BB18- 24 C9 9B F0 24 C9 88 D0
BB20- 0A 8A F0 04 CA 20 10 FC
BB28- 4C 0C FD C9 8D D0 0E 20
BB30- FE BA 20 12 B9 A9 8D A6
BB38- 45 8E B0 BD 60 C9 40 B0
BB40- 0F 09 80 20 EA BB 9D 00
BB48- 02 E8 20 1E BD 4C 0C FD
BB50- C9 8F D0 0C B1 28 20 CF
BB58- BC 20 1B FD 09 80 D0 2F
BB60- C9 91 D0 0F A5 45 86 45
BB68- 38 E5 43 AA F0 C4 20 4A
BB70- F9 F0 BF C9 98 D0 3C 20
BB78- FE BA 20 12 B9 A9 98 D0
BB80- B8 C9 89 D0 43 20 DE BC
BB88- AC 00 C0 C0 20 90 82 A6
BB90- 45 F0 11 20 EA BB 48 E8
BB98- CA BD 00 02 9D 01 02 E4
BBA0- 43 D0 F5 68 E6 45 9D 00
BBA8- 02 20 1E BD E8 20 BE BA
BBB0- 4C 85 BB C9 83 D0 CA B1
BBB8- 28 C9 E0 90 02 29 DF C9
BBC0- 40 B0 02 09 80 4C FB BC
BBC8- C9 84 D0 32 A5 45 F0 03
BBD0- 20 D6 BB 4C 0C FD E4 45
```

```
BBD8- B0 09 BD 01 02 9D 00 02
BBE0- E8 D0 F3 C6 45 A6 43 4C
BBE8- BE BA E4 44 90 05 48 20
BBF0- DD FB 68 E0 F8 90 06 68
BBF8- 68 A6 43 B0 D6 60 C9 9A
BC00- D0 03 4A 10 04 C9 86 D0
BC08- 44 85 3F 20 DE BC 4E AF
BC10- BD E8 E4 45 B0 E3 DD 00
BC18- 02 D0 F6 24 3F 10 0F A5
BC20- 43 86 43 AA E8 20 F4 FB
BC28- E4 43 90 F8 B0 0D 86 40
BC30- A6 43 20 D6 BB C6 40 E4
BC38- 40 D0 F7 20 DE BC AC 00
BC40- C0 C0 15 F0 05 DD 00 02
BC48- F0 C7 4C 17 BB C9 92 D0
BC50- 12 24 3E 10 0E 20 FE BA
BC58- A9 8D 20 F0 FD 20 F0 FD
BC60- 4C F0 B9 C9 90 D0 51 A5
BC68- 45 85 3F 20 2A BD 20 E2
BC70- BA 20 8D BC A2 00 86 43
BC78- 20 FE BA A9 A0 E4 3F B0
BC80- 06 20 EB BA E8 D0 F6 20
BC88- D5 BA 4C A2 BA 98 48 A9
BC90- 00 A8 AA 85 40 BD 00 02
BC98- C9 A0 D0 04 24 40 10 0C
BCA0- 99 00 02 C8 C9 A2 D0 04
BCA8- 45 40 85 40 E8 E4 45 90
BCB0- E4 84 45 85 3D 68 A8 60
BCB8- C9 82 D0 06 20 2A BD 4C
BCC0- 0C FD C9 8E D0 06 20 FE
BCC8- BA 4C 0C FD 4C EA BB 48
BCD0- C9 E0 90 02 29 1F 29 3F
BCD8- 09 40 91 28 68 60 A4 24
BCE0- B1 28 20 CF BC 4E AF BD
BCE8- 20 1B FD C9 93 F0 26 C9
BCF0- 81 F0 21 C9 8D F0 23 C9
BCF8- 98 F0 1F 2C AE BD 10 06
BD00- C9 C0 90 02 09 20 C9 95
BD08- D0 02 B1 28 C9 E0 90 03
BD10- 8D AF BD 60 4A 8D AE BD
BD18- B0 C4 4E AE BD 60 C9 A0
BD20- B0 53 48 29 3F 20 F0 FB
BD28- 68 60 8A F0 06 20 10 FC
BD30- CA D0 FA 60 85 38 84 39
BD38- 60 85 36 84 37 60 48 AD
BD40- 00 C0 C9 86 D0 09 4D B1
BD48- BD 8D B1 BD 8D 10 C0 68
BD50- 2C B1 BD 30 E8 C9 8D D0
BD58- 1C AD 00 C0 C9 9B F0 04
BD60- C9 93 D0 0B 8D 10 C0 AD
BD68- 00 C0 10 FB 8D 10 C0 C9
BD70- 83 F0 25 A9 8D 2C AF BD
BD78- 10 09 AD AF BD 4E AF BD
BD80- 9D 00 02 C9 8B D0 05 20
BD88- 1A FC A9 8B C9 89 D0 05
BD90- 20 F4 FB A9 89 4C F0 FD
BD98- 8D 10 C0 4C 09 B9 A0 D4
BDA0- C9 C4 C5 CC C9 D3 D4 A0
BDA8- B1 B2 B3 8D 8D 8D 00 00
BDB0- 09 00 C0 9B 40 9B C3 C1
BDB8- CC CC 20 D0 D0 CF CB 45
BDC0- C9 9B C4 7F CA 9B C2 7F
BDC8- CB 9B C1 7F CD 9B C3 7F
```

```
BDD0- BA C3 C1 CC CC AD B1 B5
BDD8- B1 0D D1 A0 D0 D2 C9 CE
BDE0- D4 D0 C5 C5 CB A8 C1 A9
BDE8- AB A8 D0 C5 C5 CB A8 C1
BDF0- AB B1 A9 A9 AA B2 B5 B6
BDF8- 0D A3 5F A4 1C A5 1F 85
BE00- 87 87 18 9B 7F B1 C3 C1
BE08- D4 C1 CC CF C7 C4 B1 0D
BE10- B3 CC CF C1 C4 20 B4 D3
BE18- C1 D6 C5 20 B6 C2 D3 C1
BE20- D6 45 B0 D6 C5 D2 C9 C6
BE28- D9 20 D2 D2 C5 D4 D5 D2
BE30- CE 20 AA D2 C5 CD A0 AA
BE38- AA AA AA AA AA AA AA AA
BE40- AA AA AA AA AA AA AA AA
BE48- AA AA AA 0D A1 D2 C5 CD
BE50- A0 AA A0 A0 A0 A0 A0 A0
BE58- A0 A0 A0 A0 A0 A0 A0 A0
BE60- A0 A0 A0 A0 2A CE CE CF
BE68- D4 D2 C1 C3 45 D4 D4 D2
BE70- C1 C3 45 D7 C2 C6 C6 C3
BE78- AE C2 C6 C6 C6 0D CC CC
BE80- C9 D3 54 B5 C2 CC CF C1
BE88- 44 C3 C3 C8 D2 A4 A8 A0
BE90- A0 29 AD C3 C1 CC 4C B7
BE98- CC CF C3 4B B8 D5 CE CC
BEA0- CF C3 4B B9 C4 C5 CC C5
BEA8- D4 45 B2 C3 C1 D4 C1 CC
BEB0- CF C7 C4 B2 0D C8 D0 CF
BEB8- CB C5 AD B1 B6 B3 B0 B2
BEC0- AC B0 BA D0 CF CB C5 AD
BEC8- B1 B6 B2 B9 B7 AC B0 BA
BED0- D0 CF CB C5 AD B1 B6 B3
BED8- B0 B4 AC B0 BA D0 CF CB
BEE0- C5 A8 AD B1 B6 B3 B0 B1
BEE8- AB D0 A9 AC B0 0D 00 C5
BEF0- 0D 00 45 00 00 A0 A0 2A
BEF8- 00 CC CF C7 C4 B2 8D 00
BF00- D3 18 90 05 A0 0D 91 48
BF08- 38 8D 81 C0 20 8D FD 4C
BF10- 66 BF 8D 81 C0 20 ED FD
BF18- 4C 66 BF 8D 81 C0 20 DA
BF20- FD 4C 66 BF 8D 81 C0 20
BF28- 8B FE 4C 66 BF 8D 81 C0
BF30- 20 95 FE 4C 66 BF 8D 81
BF38- C0 20 65 D6 20 66 BF 4C
BF40- FF E4 8D 81 C0 4C D2 D7
BF48- 8D 81 C0 4C 36 E8 20 2D
BF50- F7 4C 66 BF 8D 81 C0 20
BF58- 5D BF 4C 66 BF 6C 38 00
BF60- 8D 81 C0 20 6F BF 08 2C
BF68- 83 C0 2C 83 C0 28 60 6C
BF70- 36 00 8D 81 C0 CD 00 E0
BF78- 4C 66 BF 8D 81 C0 4C 3C
BF80- D4 8D 81 C0 6C FA FF 8D
BF88- 81 C0 6C FC FF 8D 81 C0
BF90- 6C FE FF 20 66 BF 20 81
BF98- DE 8D 81 C0 60 20 66 BF
BFA0- 20 BD DE 4C 99 BF 8D 81
BFA8- C0 AD 00 E0 D0 B8 20 66
BFB0- BF 4C BF DD 20 66
```

清楚了解檔案佔用位置

磁碟檔案地圖

張守至

APPLE的眾多週邊器材之中，要算磁碟機是最有用的了，而我們的磁碟則存有很多檔案，若果我們能夠知道磁碟上的檔案位置的話，在處理和修護磁碟的故障時便方便得多了，這是本程式的目的。

這個程式可以將磁碟檔案的地圖顯示在螢幕上或傾印到印字機去，基本上的操作很簡單。圖1所示是程式的主目錄。

```
CURRENT  CONFIGURATION                          圖1

1)  DISK  DRIVE  . . . . . . . . . . 1
2)  DISK  SLOT   . . . . . . . . . . 6
3)  OUTPUT  (C=CRT,P=PRINTER) . . . C
4)  FORMAT  (D=DECIMAL,H=HEX) . . . H
5)  PRINTER  SLOT  . . . . . . . . . 1

SELECT  (1-5)  TO  CHANGE  CONFIGURATION
<ESC>  TO  EXIT,  <RETURN>  TO  CONTINUE
```

按下1可以改變使用中的磁碟機（1或2）。

按2可以改變CONTROLLER咭所在的位置。

按3可以將輸出送到螢幕或印字機上去。

按4可以將輸出格式定為十進制或十六進制模式。

按下5可以設定印字機的位置。

按ESC鍵會離開本程式。

按RETURM則程式開始工作。

圖2是本程式的典型輸出結果。最左邊用英文字母代表相對的檔案，在地圖上大家可看見由不同檔案以相對的字母代表所佔用的位置。

本程式由兩個檔案組成，一個是BASIC寫成的驅動器，另一個是機械語言寫成的主程式。檔案名分別為DISK FILE MAP DRIVER及DISK FILE MAP ■

```
CATALOG OF DISK VOLUME 254

A- A 035 PERFECT LABEL CREATING MACHINE
B- A 011 SET PRINTER
C- A 014 DISK MAP
D- B 007 DISK.MAP.II2
E- A 014 DARWIN'S THEORY
F- A 012 LOW-RES DRAWING
G- A 005 DRAWING FILE
H- A 003 PIE CHART
I- T 002 LABEL.DATA
J- A 007 MERRY CHRISTMAS.
K- A 067 EXECUTIVE ANALYZER
L- T 001 T
M- B 008 PROGRAM EDITOR

     TRACK/SECTOR MAP FOR DISK VOLUME 254

     TRACK NUMBER
     0123456789ABCDEF0123456789ABCDEF012

S 0  *................*AACDEGKKKKK......  0
E 1  *................*AACDEGKKKKK......  1
C 2  *................*AABDEGJKKKK......  2
T 3  *................*AABDEFJKKKK......  3
O 4  *................*AABCEFJKKKK......  4
R 5  *................*AABCEFJKKKK......  5
  6  *................*AABCEFJKKKK......  6
N 7  *................*AABCEFJKKKK......  7
U 8  *................*AABCEFJKKKK..M...  8
M 9  *................*AABCEFIKKKK..M...  9
B A  *................*AABCEFIKKKK..M...  A
E B  *................*AABCEFHKKKK..M...  B
R C  *................*AABCEFHKKKK..M...  C
  D  *................*AAACDFHKKKK..M...  D
  E  *................*AAACDFGKKKK..M...  E
  F  *................*AAACDEGKKKKKLM...  F

     0123456789ABCDEF0123456789ABCDEF012
      *= SYSTEM SECTOR   .=EMPTY SECTOR      圖2
```

```
REM   ::::::::::::::::::
REM     DISK FILE MAP
REM     BINARAY  FILE
REM      *8000.8549
REM   ::::::::::::::::::

8000- 4C 29 81 4C 58 82 4C A9
8008- 82 4C 95 83 4C 0F 83 4C
8010- 09 85 B0 B1 B2 B3 B4 B5
8018- B6 B7 B8 B9 C1 C2 C3 C4
8020- C5 C6 B0 B1 B2 B3 B4 B5
8028- B6 B7 B8 B9 B0 B1 B2 B3
8030- B4 B5 D3 C5 C3 D4 CF D2
8038- A0 CE D5 CD C2 C5 D2 A0
8040- A0 A0 C1 C2 C3 C4 C5 C6
8048- C7 C8 C9 CA CB CC CD CE
8050- CF D0 D1 D2 D3 D4 D5 D6
8058- D7 D8 D9 DA B0 B1 B2 B3
8060- B4 B5 B6 B7 B8 B9 01 02
8068- 03 04 05 06 07 08 09 0A
8070- 0B 0C 0D 0E 0F 10 11 12
8078- 13 14 15 16 17 18 19 1A
8080- 30 31 32 33 34 35 36 37
8088- 38 39 41 42 43 44 45 46
8090- 47 48 49 4A 4B 4C 4D 4E
```

磁碟檔案地圖

```
8098- 4F 50 51 52 53 54 55 56
80A0- 57 58 59 5A 70 71 72 73
80A8- 74 75 76 C1 C2 C3 C4 C5
80B0- C6 C7 C8 C9 CA CB CC CD
80B8- CE CF D0 D1 D2 D3 D4 D5
80C0- D6 D7 D8 D9 DA B0 B1 B2
80C8- B3 B4 B5 B6 B7 B8 B9 E1
80D0- E2 E3 E4 E5 E6 E7 E8 E9
80D8- EA EB EC ED EE EF F0 F1
80E0- F2 F3 F4 F5 F6 F7 F8 F9
80E8- FA A1 A3 A4 A5 A6 A7 A8
80F0- A9 AB AD C0 BC BE DE BF
80F8- AF DD DB DC FB FD FC BD
8100- BA BB D0 D1 D2 D3 D4 D5
8108- D6 D7 D8 D9 DA B0 B1 B2
8110- B3 B4 B5 B6 01 60 02 00
8118- 11 00 25 81 48 85 00 00
8120- 01 00 00 60 01 00 01 EF
8128- D8 20 58 FC 20 DC AB A9
8130- 00 85 07 85 0C A9 FF 85
8138- 05 8D F9 B5 20 F7 AF A9
8140- 15 8D 9D B3 20 2E 85 20
8148- EA 84 C3 C1 D4 C1 CC CF
8150- C7 A0 CF C6 A0 C4 C9 D3
8158- CB A0 D6 CF CC D5 CD C5
8160- A0 00 A2 00 86 45 AD F6
8168- B7 85 44 20 42 AE 20 2E
8170- 85 20 2E 85 18 20 11 B0
8178- B0 68 A2 00 8E 9C B3 BD
8180- C6 B4 F0 5E 30 55 20 ED
8188- 81 A0 A0 BD C8 B4 10 02
8190- A0 AA 98 20 ED FD BD C8
8198- B4 29 7F A0 07 0A 0A B0
81A0- 03 88 D0 FA B9 A7 B3 20
81A8- ED FD A9 A0 20 ED FD BD
81B0- E7 B4 85 44 BD E8 B4 85
81B8- 45 20 42 AE A9 A0 20 ED
81C0- FD E8 E8 E8 A0 1D BD C6
81C8- B4 09 80 C9 A0 B0 02 A9
81D0- DF 20 ED FD E8 88 10 EE
81D8- 20 2E 85 20 30 B2 90 9C
81E0- B0 93 A9 FF A4 07 99 84
81E8- 88 99 85 88 60 E6 05 20
81F0- 11 82 85 06 A4 07 BD C6
81F8- B4 99 84 88 C8 BD C7 B4
8200- 99 84 88 C8 84 07 A5 06
8208- 20 ED FD A9 AD 20 ED FD
8210- 60 A4 05 24 19 30 04 B9
8218- 42 80 60 B9 AB 80 60 20
8220- 29 82 A5 06 A0 00 91 00
8228- 60 A9 00 85 03 A5 08 0A
8230- 26 03 0A 26 03 0A 26 03
8238- 0A 26 03 85 02 18 A9 49
8240- 65 02 85 00 A9 86 65 03
8248- 85 01 18 A5 00 65 09 85
8250- 00 A5 01 69 00 85 01 60
8258- A2 00 86 08 86 0B A9 10
8260- 85 09 A6 0B BD F3 B3 20
8268- 83 82 A6 0B BD F4 B3 20
8270- 83 82 E6 0B E6 0B E6 0B
8278- E6 0B E6 08 A5 08 C9 23
8280- 90 DC 60 85 04 A2 09 CA
8288- D0 01 60 C6 09 06 04 B0
8290- 0C 20 29 82 A9 AA A0 00
8298- 91 00 4C 87 82 20 29 82
82A0- A9 AE A0 00 91 00 4C 87
82A8- 82 A0 00 84 07 88 84 05
82B0- E6 05 20 11 82 85 06 A4
82B8- 07 B9 84 88 10 01 60 8D
82C0- 18 81 85 08 C8 B9 84 88
82C8- 8D 19 81 85 09 C8 84 07
82D0- 20 1F 82 A0 14 A9 81 20
82D8- D9 03 A2 0C BD 48 85 85
82E0- 08 E8 E0 01 F0 11 BD 48
82E8- 85 85 09 E8 18 65 08 F0
82F0- 06 20 1F 82 4C DC 82 A2
82F8- 01 BD 48 85 F0 B2 8D 18
8300- 81 85 08 E8 BD 48 85 8D
8308- 19 81 85 09 4C D0 82 20
8310- 58 FC 20 8A 84 20 EA 84
8318- A0 A0 D4 D2 C1 C3 CB A0
8320- CE D5 CD C2 C5 D2 8D 00
8328- A9 A0 20 ED FD A9 3F 85
8330- 32 20 28 84 A9 00 85 09
8338- A9 00 85 08 A4 09 B9 32
8340- 80 20 ED FD A5 1A D0 05
8348- B9 22 80 D0 03 B9 12 80
8350- 29 3F 20 ED FD 20 29 82
8358- A0 00 B1 00 20 ED FD E6
8360- 08 A5 08 C9 23 90 EE A4
8368- 09 A5 1A D0 05 B9 22 80
8370- D0 03 B9 12 80 29 3F 20
8378- ED FD 20 8E FD E6 09 A5
8380- 09 C9 10 90 B3 A9 A0 20
8388- ED FD A9 3F 85 32 20 28
8390- 84 20 C3 84 60 A9 04 85
8398- 24 20 8A 84 20 EA 84 A0
83A0- A0 A0 A0 A0 D4 D2 C1 C3
83A8- CB A0 CE D5 CD C2 C5 D2
83B0- 8D 00 A9 04 85 24 20 28
83B8- 84 20 8E FD A9 00 85 09
83C0- A9 00 85 08 A4 09 B9 32
83C8- 80 20 ED FD A9 A0 20 ED
83D0- FD A5 1A D0 05 B9 22 80
83D8- D0 03 B9 12 80 20 ED FD
83E0- A9 05 85 24 20 29 82 A0
83E8- 00 B1 00 20 ED FD E6 08
83F0- A5 08 C9 23 90 EE A9 2A
83F8- 85 24 A4 09 A5 1A D0 05
8400- B9 22 80 D0 03 B9 12 80
```

```
8408- 20 ED FD 20 8E FD E6 09
8410- A5 09 C9 10 90 AA 20 8E
8418- FD A9 04 85 24 20 28 84
8420- A9 04 85 24 20 C3 84 60
8428- A5 1A F0 2F 20 EA 84 A0
8430- B0 B1 B2 B3 B4 B5 B6 B7
8438- B8 B9 C1 C2 C3 C4 C5 C6
8440- B0 B1 B2 B3 B4 B5 B6 B7
8448- B8 B9 C1 C2 C3 C4 C5 C6
8450- B0 B1 B2 A0 8D 00 A9 FF
8458- 85 32 60 20 EA 84 A0 B0
8460- B1 B2 B3 B4 B5 B6 B7 B8
8468- B9 B0 B1 B2 B3 B4 B5 B6
8470- B7 B8 B9 B0 B1 B2 B3 B4
8478- B5 B6 B7 B8 B9 B0 B1 B2
8480- B3 B4 A0 8D 00 A9 FF 85
8488- 32 60 20 EA 84 A0 D4 D2
8490- C1 C3 CB AF D3 C5 C3 D4
8498- CF D2 A0 CD C1 D0 A0 C6
84A0- CF D2 A0 C4 C9 D3 CB A0
84A8- D6 CF CC D5 CD C5 A0 00
84B0- A2 00 86 45 AD F6 B7 85
84B8- 44 20 42 AE 20 8E FD 20
84C0- 8E FD 60 20 EA 84 A0 A0
84C8- AA BD A0 D3 D9 D3 D4 C5
84D0- CD A0 D3 C5 C3 D4 CF D2
84D8- A0 A0 AE BD C5 CD D0 D4
84E0- D9 A0 D3 C5 C3 D4 CF D2
84E8- 00 60 68 85 00 68 85 01
84F0- E6 00 D0 02 E6 01 A2 00
84F8- A1 00 F0 06 20 ED FD 18
8500- 90 EE A5 01 48 A5 00 48
8508- 60 A2 30 A0 0A 20 18 85
8510- A2 18 A0 14 20 18 85 60
8518- 86 00 84 01 AD 30 C0 88
8520- D0 04 C6 01 F0 07 CA D0
8528- F6 A6 00 D0 EF 60 20 8E
8530- FD E6 0C A5 19 D0 10 CE
8538- 9D B3 D0 0B 20 0C FD 20
8540- 58 FC A9 15 8D 9D B3 60
8548- 09 FF
```

```
10  REM  :::::::::::::::::
20  REM    DISK FILE MAP
30  REM         DRIVER
40  REM    COPYRIGHT (C)
50  REM      1984     BY
60  REM    COMPUTING AGE
70  REM  :::::::::::::::::
130 B$ =  CHR$ (7)  +  CHR$ (7) +
     CHR$ (7)
140 D$ =  CHR$ (13)  +  CHR$ (4)
150 REM
160 REM  LOAD ASSEMBLY LANGUAGE
170 REM  SUBROUTINES
180 REM
190 PRINT D$"BLOAD DISK.MAP.II2"

200 REM
210 GOSUB 610: REM INTRODUCTION
220 GOSUB 890: REM CONFIGURATION

230 REM
240 REM   MAIN LOOP
250 REM
260 IF OD$ = "P" THEN  PRINT D$;
    "PR#"PS%; CHR$ (13) +  CHR$
    (9) + "80N"
270 POKE 46528,DR%: REM  $B5C0
280 POKE 46529,DS%: REM  $B5C1
290 REM
300 REM  POKE DISK DRIVE AND
310 REM  SLOT INTO IOCB
320 REM
330 POKE IOCB + 1,DS% * 16
340 POKE IOCB + 15,DS% * 16
350 POKE IOCB + 2,DR%
360 REM
370 CALL 32768: REM  CATALOG
380 REM
390 IF OD$ = "C" THEN  GOSUB 570
    : HOME : PRINT : PRINT "LOAD
    ING THE TRACK SECTOR MAP": PRINT
    : FLASH : PRINT " PLEASE WAI
    T ": NORMAL
400 REM
410 CALL 32771: REM  CLEAR MAP
420 CALL 32774: REM  LOAD MAP
430 REM
440 REM   DISPLAY THE MAP
450 REM
460 IF OD$ = "C" THEN  CALL 3278
    0: PRINT : GOSUB 570: GOTO 2
    20
470 REM
480 REM   PRINT THE MAP
490 REM
500 IF  PEEK (12) > 33 THEN  PRINT
     CHR$ (12)
510 PRINT : CALL 32777: PRINT  CHR$
    (12);
520 PRINT D$"PR#0": GOTO 220
530 REM
540 END
550 REM MISC ROUTINES
560 FOR K = 1 TO 1000: NEXT K: RETURN

570 POKE 49168,0: PRINT : PRINT
    " PRESS <RETURN> TO CONTINUE
    ";: CALL BLEEP
580 GET Z$: IF  ASC (Z$) <  > 13
     AND  ASC (Z$) <  > 141 THEN
     GOTO 580
590 RETURN
```

```
600  REM  INTRODUCTION
610 IOCB = 33044: REM $8114
620 DR% =  PEEK (43624): REM   CU
    RRENT DISK DRIVE
630 DS% = 6: REM   DEFAULT DISK
     SLOT
640 OD$ = "C": REM   DEFAULT OUTP
    UT TO CRT
650 OF$ = "H": REM   DEFAULT HEX
    OUTPUT FORMAT
660 PS% = 1: REM   DEFAULT PRIN
    TER SLOT
670 BLEEP = 32783
680 PF = 25:DF = 26
690  POKE PF,0: POKE DF,128
700  POKE 46528,DR%: REM  $B5C0
710  POKE 46529,DS%: REM  $B5C1
720  TEXT : HOME : VTAB 22: PRINT
    "** COPYRIGHT 1984 BY CIMPUT
    ING  AGE   **": VTAB 1
730  HTAB 13: INVERSE : PRINT "DI
    SK MAP II": NORMAL
740  PRINT : PRINT : PRINT  TAB(
    12)"BY COMPUHOUSE": PRINT : PRINT

750  PRINT "   THIS PROGRAM WILL
    PRINT A MAP WHICH"
760  PRINT " WILL INDICATE THE AC
    TUAL TRACK/SECTOR"
770  PRINT " LOCATION OF EACH FIL
    E ON A  DISKETTE."
780  PRINT " A CATALOG  OF THE  D
    ISKETTE  WILL  BE"
790  PRINT " PRINTED FIRST.  BEFO
    RE EACH FILE NAME"
800  PRINT " IS A CHARACTER WHICH
     WILL BE  USED TO"
810  PRINT " INDICATE THE LOCATIO
    N OF  THE FILE ON"
820  PRINT " THE DISK MAP.   IN A
    DDITION, ANY CON-"
830  PRINT " TROL CHARACTERS  EMB
    EDDED IN THE FILE"
840  PRINT " NAME WILL BE REPLACE
    D WITH AN  UNDER-"
850  PRINT " LINE '"; CHR$ (223)"
    '."
860  GOSUB 570
870  RETURN
880  REM SET CONFIGURATION
890  HOME
900  VTAB 1: HTAB 13: INVERSE : PRINT
    "DISK MAP II": NORMAL : PRINT

910  PRINT "CURRENT CONFIGURATION
    ": PRINT
920  PRINT "1) DISK DRIVE . . . .
     . . . . . "DR%
930  PRINT "2) DISK SLOT  . . . .
     . . . . . "DS%
940  PRINT "3) OUTPUT (C=CRT,P=PR
    INTER) . . "OD$
950  PRINT "4) FORMAT (D=DECIMAL,
    H=HEX) . . "OF$
960  PRINT "5) PRINTER SLOT . . .
     . . . . . "PS%
970  PRINT : PRINT
980  PRINT "SELECT (1-5) TO CHANG
    E CONFIGURATION"
990  VTAB 13: PRINT "<ESC> TO EXI
    T, <RETURN> TO CONTINUE ";: CALL
    BLEEP: GET Z$
1000  IF  ASC (Z$) = 13 OR  ASC (
    Z$) = 141 THEN  PRINT : RETURN

1010  IF  ASC (Z$) = 27 THEN  HOME
    : END : REM  ESC
1020  IF Z$ = "1" THEN  GOSUB 109
    0: GOTO 900
1030  IF Z$ = "2" THEN  GOSUB 113
    0: GOTO 890
1040  IF Z$ = "3" THEN  GOSUB 121
    0: GOTO 900
1050  IF Z$ = "4" THEN  GOSUB 124
    0: GOTO 900
1060  IF Z$ = "5" THEN  GOSUB 127
    0: GOTO 890
1070  PRINT : GOTO 990
1080  REM TOGGLE DISK DRIVE
1090 DR% = (DR% = 2) + 2 * (DR% =
    1) + (DR% <  > 1) * (DR% <  >
    2)
1100  POKE 43624,DR%: REM $AA68
1110  RETURN
1120  REM ENTER DISK SLOT
1130  HOME : PRINT : PRINT
1140  PRINT "INPUT THE DISK DRIVE
    SLOT NUMBER"
1150  PRINT " (1 THROUGH 7) ";: GET
    S$:DS% =  VAL (S$): PRINT S$

1160  IF DS% >  = 1 AND DS% <  =
    7 THEN  RETURN
1170  PRINT : PRINT : FLASH : PRINT
    B$" INCORRECT DISK DRIVE SLO
    T NUMBER ": NORMAL
1180  GOSUB 560
1190  GOTO 1130
1200  REM TOGGLE OUTPUT
1210  IF OD$ = "C" THEN OD$ = "P"
    : POKE PF,128: RETURN
1220  IF OD$ = "P" THEN OD$ = "C"
    : POKE PF,0: RETURN
1230  REM TOGGLE FORMAT
1240  IF OF$ = "D" THEN OF$ = "H"
    : POKE DF,128: RETURN
1250  IF OF$ = "H" THEN OF$ = "D"
    : POKE DF,0: RETURN
1260  REM GET PRINTER SLOT
1270  HOME : PRINT : PRINT
1280  PRINT "INPUT THE PRINTER SL
    OT NUMBER"
1290  PRINT " (1 THROUGH 7) ";: GET
    S$:PS% =  VAL (S$): PRINT S$

1300  IF PS% >  = 1 AND PS% <  =
    7 THEN  RETURN
1310  PRINT : PRINT : FLASH : PRINT
    B$" INCORRECT PRINTER SLOT N
    UMBER ": NORMAL
1320  GOSUB 560
1330  GOTO 1270
```

保護程式技巧之一搬移VTOC法

麥少文

談到程式的保護，可說是與程式的拷貝成了永遠的敵人，這場延綿不絕的戰爭，好像是永沒有完的一日。

然而，在理論上，沒有一個程式是不能被拷貝的。而保護技巧的作用，也只是基於時間性問題而已；至少要花破解者一段時間的追尋及研究，始能如願。

言歸正傳，我們現在談的是將磁碟中的ＶＴＯＣ（內容冊），搬移至23磁軌（一個鮮為人注意的地方），從而達到保護的目的，因為通常拷貝程式只拷貝第０磁軌至22磁軌（16進數，即10進數的第35磁軌），除非使用者特別指示拷貝工具連＄23磁軌也一起拷貝，但也不是每一樣拷貝工具程式都能拷貝23磁軌。所以此種保護方法，至少可在朋友面前神秘一番，而能否破解，則視乎對手的道行有多高深了。

在正常的磁碟中，正常的DOS被BOOT入RAM內時，會命令磁碟機的讀寫頭移至11磁軌（16進數，10進數是17磁軌），與讀取０磁區的VTOC，找尋CATALOG SECTOR，從中再讀取要執行的FILE的TRACK／SECTOR LIST所在的SECTOR ，最後才跳至實際存放資料的某磁軌中的某磁區去讀取資料及執行。若讀者對上述之觀念不大清楚的話，可參考市面上有關ＤＯＳ的書（如磁碟系統詳論一書），自有清楚交代。

我們要注意的，只集中在VTOC上，因為它是DOS 第一步去讀的地方；也是讀寫頭的必經之路（在儲存多個FIL—E的程式磁碟而言，若是某些特別處理的磁碟，只儲存一個程式在磁碟中而甚至沒有VTOC及CATALOG SECTOR的磁碟，則作別論）。

首先，我們要INIT 一張具有23磁軌或27磁軌（十進位數是40磁軌）的軌碟，若是把ＶＴＯＣ搬至40磁軌的話，則保護能力會大大的提高。原因有三：第一，因為有一部份的磁碟機，根本不能讀寫40磁軌的。第二，有些磁碟的質素也不能劃分有效的40條磁軌。第三，若有人想拷貝40條磁軌，所用的拷貝工具程式未必能拷貝40磁軌（據筆者所知，大多數的拷貝工具程式都不能拷貝40磁軌）。

操作細則：

㈠BOOT普通的HELLO 程式，然後按５，跳出程式回到APPLESOFT BASIC的狀態（此時正常的DOS 已存在RAM內了）。

㈡在APPLESOFT BASIC的PROMPT CURSOR〕■前，按CALL—151（進入監督系統），然後按RETURN ，＊號隨著出現。

㈢在＊前按入BEFE：24（修改DOS），隨著RETURN，又出現＊號，按CONTROL—C，再按RETURN ，回復〕■的BASIC PROMPT。或在＊號前按入3D0G ，再RE—TURN，效果一樣可以將＊號轉為〕■。在此處輸入的值24，是準備規劃一隻36軌的磁碟，若要40軌，則需按入28。也就是說，在ＢＥＦＥ位址的數值，就是磁碟軌數的數目加１。筆者曾經多次實驗，ＢＥＦＥ的數值，至多可按入29，亦即能劃出一隻41軌的磁碟，再增大數值，效果一樣。讀者可自行試試，自己的磁碟機能否規劃40或41軌的磁碟。（注意：輸入的數是十六進，得出的軌數是十進。）

㈣承上所述，從＊號轉回APPLFSOFT 的〕■後，便可放入一隻空白的磁碟，並按入INIT HELLO的命令，隨著RETURN 。此時，經剛才步驟修改過的ＤＯＳ，便會規劃一隻36或40軌的磁碟（視乎讀者在BEFE中輸入了24或28），若讀者在碟機旁，細心聆聽，可聽到在規劃時，一共發出了36或40聲軋軋聲，即表示碟機正在規劃每一條軌所發出的聲音，我們準備要一張40軌的磁碟，當然可以聽到40聲的軋軋聲罷。讀者可以此張查驗自己剛才輸入於BEFE 的數值有否錯誤。

㈤至此，我們可得到一隻40軌的空碟。但在搬移VTOC之前，我們需要用到一些工具程式，如DISK ZAP或INSPE—CTOR之類。此類工具程式多藏於COPY PROGRAM 之內。如LOCKSMITH 中有INSPECTOR 的程式。筆者是選用了INSPECTOR 。首先，將一隻未經保護的磁碟的程式全部拷貝到剛才做好的40軌磁碟上。然後所做的工作，全部於這隻40軌的磁碟上發生。

㈥首先要載入INSPECTOR 程式，只見螢幕上有兩行文字，其他空白（如圖一）。

```
TRACK 00  SECTOR 0      SLOT 6   DRIVE 1
          BUFFER 0800   DOS 16   VER 6.1
=========================================
```

圖(1)

此時，按入Ｔ（輸入TRACK的數目），再按11，然後RETURN，同樣按入Ｓ（輸入SECTOR之數目），再按０，隨著RETURN 。於此，TRACK上的數目應為11，而SEC—TOR的數目應為０。（TRACK 11，SECTOR 0 也正是所有正常磁碟的ＶＴＯＣ所在。）

㈦按Ｒ（READ入VTOC），畫面隨即改變，如圖二。圖中可見大部份軌都是空的（以FFFF 表示）。只是TRACK ＄11整條都佔滿了（以０００００於BYTE ＄7C及BYTE ＄7D 表示），此乃VTOC內的BIT MAP ，讀者可自行參考其他書，而不在本文範圍。我們要做的是，於出現圖二畫面後，隨即按Ｔ，再按入28，及RETURN ，最後按CONT—ROL—W（把資料寫到41TRACK去）。注意：前面做規劃磁碟時，在BEFE要輸入29，始有41條軌，否則此處按Ｔ後只能按入27（40TRACK）或按入23（搬至36TRACK）。見圖三，已將VTOC搬至41軌（即十六進數的28TRACK

），所以見到TRACK的數目是28，而VTOC的內容和圖二一樣。

(八)步驟七已將VTOC搬至41 TRACK，若這樣以為完工就錯了。因為本來的VTOC仍然留在TRACK 11，SECTOR 0中。所以要再應用上述的技巧，把一空白的SECTOR（如TRACK $ 28內其他SECTOR，除0 SECTOR外）讀入（按R）然後再CONTROL—W，把空的SECTOR 寫入TRACK $ 11，SECTOR 0 中，如能洗去原有的VTOC。（當然，在讀入一空SECTOR後，要再按T及11，按RET—URN，及按S然後輸入0，按RETURN。更正了TRAC—K 11 SECTOR 0後，才按CONTROL—W，把空資料寫進去。）

(九)至此，若以這碟BOOT 機，碟內程式不能執行且出現DISK FULL字樣。因為碟內之DOS仍以為VTOC在—TRACK 11 SECTOR 0內（但已是全部0的空SECTOR，在BIT MAP中兩個0 0的BYTE 是代表整條軌用滿了，現在整個SECTOR都是0，所以DOS便以為整張碟都用滿了。所以，我們要修改DOS，使其跳至41軌處找已搬了家的正確VTOC。方法是按入T及01後RETURN；及按入S及B再RETURN，即選定TRACK 1 SECTOR B，然後讀入（按R），出現如圖四。可以見到在BYTE $ 01 的數值仍是11（代表VTOC所在的TRACK數字）。我們應將以改為28才對。方法是按E01按SPACE BAR 按入28跟著RE—TURN，然後CONTROL—W寫回去。

(十)現在可說是大功告成了。重新BOOT機，此已做了手脚的碟，照RUN如是，沒有問題。若此碟是裝了多個—FILE 的，則可用此碟BOOT機後，用CATALOG的命令已列出各FILE來。但若用其他碟BOOT機，再CATA—LOG此碟，出現的只是DISK VOLUME 254，其他什麽也沒有。因為正常的DOS找不到這隻特別碟（搬了VTOC）的VTOC，所以便不能找到CATALOG SECTOR。若想戲弄一下別人，在這隻特別碟的TRACK 11 SECTOR 0（本來是VTOC的地方，現在已完全是0）中的BYTE 01 存入FF值（方法是讀入了TRACK 11 SECTOR 0 的空資料後，按E01按SPACE BAR再按FF及RETURN，用CON—TROL—W 把它寫回）。此時，若有別人在正常的DOS去CATALOG 此碟，後果是使磁碟機受苦（不斷發出軋軋作响的聲音），因為是剛才存入的FF值作怪，在TRACK 11 SECTOR 0的BYTE 01中，本來是存放CATALOG TRACK的地方，現在存入FF，當然會使讀寫頭去上下拉動，怎樣找也找不到了。

至此，本文已告一段落，希望各位成功。■

```
TRACK 11   SECTOR 0        SLOT 6   DRIVE 1
           BUFFER 0800     DOS 16      CAC3
===========================================
    0 1 2 3  4 5 6 7  8 9 A B  C D E F
  -----------------------------------------
00- 04110F03 0000FE00 00000000 00000000
10- 00000000 00000000 00000000 00000000
20- 00000000 0000007A 00000000 00000000
30- 12010000 23100001 00000000 00000000
40- 00000000 FFFF0000 FFFF0000 FFFF0000
50- FFFF0000 FFFF0000 FFFF0000 FFFF0000
60- FFFF0000 FFFF0000 FFFF0000 FFFF0000
70- FFFF0000 FFFF0000 FFFF0000 00000000
80- 0FFF0000 FFFF0000 FFFF0000 FFFF0000
90- FFFF0000 FFFF0000 FFFF0000 FFFF0000
A0- FFFF0000 FFFF0000 FFFF0000 FFFF0000
B0- FFFF0000 FFFF0000 FFFF0000 FFFF0000
C0- FFFF0000 00000000 00000000 00000000
D0- 00000000 00000000 00000000 00000000
E0- 00000000 00000000 00000000 00000000
F0- 00000000 00000000 00000000 00000000
```

圖(2)

```
TRACK 28   SECTOR 0        SLOT 6   DRIVE 1
           BUFFER 0800     DOS 16      CAC3
===========================================
    0 1 2 3  4 5 6 7  8 9 A B  C D E F
  -----------------------------------------
00- 04110F03 0000FE00 00000000 00000000
10- 00000000 00000000 00000000 00000000
20- 00000000 0000007A 00000000 00000000
30- 12010000 23100001 00000000 00000000
40- 00000000 FFFF0000 FFFF0000 FFFF0000
50- FFFF0000 FFFF0000 FFFF0000 FFFF0000
60- FFFF0000 FFFF0000 FFFF0000 FFFF0000
70- FFFF0000 FFFF0000 FFFF0000 00000000
80- 0FFF0000 FFFF0000 FFFF0000 FFFF0000
90- FFFF0000 FFFF0000 FFFF0000 FFFF0000
A0- FFFF0000 FFFF0000 FFFF0000 FFFF0000
B0- FFFF0000 FFFF0000 FFFF0000 FFFF0000
C0- FFFF0000 00000000 00000000 00000000
D0- 00000000 00000000 00000000 00000000
E0- 00000000 00000000 00000000 00000000
F0- 00000000 00000000 00000000 00000000
```

圖(3)

```
TRACK 01   SECTOR B        SLOT 6   DRIVE 1
           BUFFER 0800     DOS 16      C396
===========================================
    0 1 2 3  4 5 6 7  8 9 A B  C D E F
  -----------------------------------------
00- A9118DFA B560201D AF2034AF 20C3B2A9
10- 022DD5B5 F02120F7 AFA90018 2011B038
20- CED8B5D0 F7AED9B5 ADEEB59D E7B4ADEF
30- B59DE8B4 2037B04C 7FB32028 ABADF6B5
40- 302BADBD B58542AD BEB58543 AE9CB320
50- 1CB22037 B04C7FB3 ADBCB5C9 05B00B0A
60- AABDE6AA 48BDE5AA 48604C67 B34C7BB3
70- ADF6B530 F8ADBCB5 C905B0EE 0AAABDF2
80- AA48BDF1 AA486020 00B320A8 AC8DC3B5
90- 4C7FB320 00B320B5 B120A8AC 4820A2B1
A0- A0006891 424C96AC 20B6B0B0 0BB14248
B0- 205BB120 94B16860 4C6FB320 00B3ADC3
C0- B520DAAC 4C7FB320 00B320A2 B1A000B1
```

圖(4)

最具威力的拷貝工具之一

李德華

LOCKSMITH 5.0使用方法探討

LOCKSMITH 5・0 是目前市面上較具威力的拷貝程式之一。但相信很多人除了拷貝外，便忽略了LOCKSMITH 5・0 的其他功能，現在筆者詳盡介紹其功能：

當你把LOCKSMITH 磁碟放入第一號磁碟機內之後，再把電腦的電源開關開啓，電腦螢光屏上便會出現一些文字，如：

LOCKSMITH 5・0 BOOTING—PLEASE WAIT／TO LOAD SOLT O RAM CARD，PRESS "L" WHILE BOOTING。

上列的文字是以LOCKSMITH 5・0 爲根據，那些文字指示操作者按下L鍵，使INT BASIC 載入到插在SOLT 0上的RAM 咭上。如沒有RAM 咭的用者，請不要按 "L" 鍵。

跟着，螢光屏上會出現一些資料，如LOCKSMITH 的版權聲明，出品公司和磁碟編號等。之後，我們按下SPACE 鍵，螢光屏上就會出現一個「主MENU」，其中包括LOCK—SMITH 的「狀態欄」(在「主MENU」上方的四列空白位置) 及一系列的選擇項目，以下逐一解釋：

一、〔B〕BACKUP／COPY DISK —故名思義就是複製(COPY) 磁碟。當操作者選擇了〔B〕以後，LOCKSMITH 會要求操作者把LOCKSMITH 磁碟放入磁碟機內，然後按SPACE 鍵。螢光屏將會出現以下訊息：PRESS ESC TO ABORT／DRIVE—ORIGINAL：1 ，這是詢問用者主磁碟所在的磁碟機編號，或按下RETURN 鍵使用LOCKSMITH 的預設值(1)。

跟着，LOCKSMITH會詢問用者，COPY DISK 所在的磁碟機的編號，如得一個磁碟機的用者，這裏應設置爲(1)。之後，LOCKSMITH 會要求輸入磁軌起點、磁軌終點及增軌值 (最少可設置爲0・75)，還會詢問用者是否需要同步(SYNCHRONIZE)，因有些磁碟是使用同步的保護措施，但現今市面上較少人採用這種保護措施。跟着，LOCKSMITH 又會詢問用者是否需要保留磁軌的長度，因爲採用半位元組計算的保護措施的磁碟，在使用時會檢查某些磁軌的半位元組數目，如果數目不符或相差太大，磁碟便無法操作。

這種保留磁軌長度的功能，可以使COPY DISK的磁軌長度與ORIGINAL DISK 相同，因此使複製工作更易進行。這種形態共有三種：一、N —不使用半位元組計算；二、A —由LOCKSMITH 自動調整「磁軌長度」；三、M —由用者自行設定「磁軌長度」。通常會選擇N或A。

當用者選擇了後，LOCKSMITH 又會詢問用者，是否需要AUTO ERROR RETRY，這項功能通常選擇 "Y"。

當用者完成所有步驟後，LOCKSMITH 會指示用者把ORIGINAL 碟及COPY DISK 放進磁碟機 (如得一個磁碟機，只需放入ORIGINAL 碟)，並按SPACE 鍵。跟着LO—CKSMITH 會從主磁碟讀取資料並加以分析，再而寫落COPY DISK 裏，複製期間，往往會發生一些錯誤，LOCK—SMITH 會以6種數目字代表不同的錯誤。

0—沒有錯誤；1—磁軌內沒有資料或找到一些無意義的資料；2—無法找到磁軌的起點；4—比較第一次和第二次的磁軌映象 (TRACK IMAGE) 後，發現錯誤；5—超過了規限時間，而仍然無法完成「半位元組計算」；8—寫後審核發現錯誤。

二、〔P〕PARAMETER CHANGE

這項功能容改用者修改LOCKSMITH 的「複製參數」，當選擇這項功能後，螢光屏上會出現一系列參數號碼和參數值，用者可以用←、→、I、M 四個鍵控制游標的上、下、左、右移動，而，、・可以快速上捲和下捲螢幕，當游標移到需要修改的參數後，按C (CHANGE)鍵來輸入二位數目字，按RETURN 離開或按SPACE 鍵繼續修改參數。

三、〔／〕CLEAN STATUS

這項功能允許用者把LOCKSMITH 重置。

四、〔T〕TEXT EDITOR

這項功能是LOCKSMITH 程式中最實用的一個工具程式。而這裏共包括了十一項功能。現在把所有功能逐一解釋。

〔L〕LOAD FILE

這項功能容許LOAD一個FILE 入LOCKSMITH裏。

〔S〕SAVE FILE

這項功能容許把剛才LOAD 的FILE SAVE 落已FOR—MAT 的磁碟。

〔C〕CATALOG

相信不用解釋，這項功能容許用者把目錄列出來。

〔1〕DRIVE 1

選擇磁碟機一號。

〔2〕DRIVE 2

選擇磁碟機2號。

〔B〕BACKUP／COPY DISK

這功能和起先說的BACKUP／COPY 是同樣的東西。

〔A〕APPEND FILE

從磁碟讀取檔案，再加以分析。

〔D〕DELETE FILE

刪除檔案。

〔E〕ENTER EDITOR

容許用者用LOCKSMITH 的程序語言 (LPL)修改參數，這種功能於上期已詳盡解釋。

〔P〕DISK PATCH

允許用者把剛才輸入的「修改參數程式」進行修改。

〔N〕NEW FILE

把剛才輸入的FILE，全部清除。

〔X〕SYNTEX CHECK

把剛才輸入的「修改參數程式」進行VERTIFY。

〔Q〕QUICK SCAN DISK

LOCKSMITH 5.0使用方法探討

選擇這項功能，則它可以幫助你查出那些磁軌包含有資料，以便進行拷貝。QUICK SCAN 的工作形式是以高解像圖形顯示。

程式會要求你輸入磁碟機編號，起始軌，結束軌，軌增量。在輸入上述數據後，你即可指令QUICKSCAN 開始進行工作。

〔U〕16 SCTR UTILITIES

這項功能，共有五個副程式，現簡單介紹。

〔V〕16 SECTOR FAST DISK VERTIFY

這項功能容許用者檢查沒有保護的程式，選擇這項功能後，LOCKSMITH要求用者放入LOCKSMITH 磁碟及按一SPACE 鍵。跟着螢幕會出現詢問磁碟機編號的問題，當選擇了磁碟機的編號後，電腦會請你放入待測試的磁碟及按鍵。LOCKSMITH 便會以・代表核對無誤，而*代表有錯誤。

〔B〕16 SECTOR FAST DISK BACKUP

這項功能是LOCKSMITH 另一種的COPY 功能，它比先前的快得多，用這項功能，首先要放入LOCKSMITH的磁碟及按SPACE 鍵。待資料進入電腦後，便輸入主磁碟的磁碟機編號及複製碟所屬的磁碟機的編號，跟着便出現顯示SECTOR及TRACK的圖表，而左上方有〝[〞的地方，可以輸入〝V〞來偵查COPY時的準確性。如無問題，可按SPACE BAR 開始複製，48K的APPLE Ⅱ只要READ／WRITE 4次便能複製完畢，而64K只要READ／WRITE 3次便完成，128K只需1次READ／WRITE便拷貝完畢。

〔F〕16 SECTOR FORMAT

這項功能容許用者把一些空磁碟進行格式化。〝F〞代表FORMAT成功。

〔C〕16 SECTOR COMPARE

這項功能容許用者將磁碟的資料比較，但是會破壞磁碟上的資料。

〔S〕16 SECTOR SYNC SIGNATURE

這項功能將所有磁碟讀出的資料，以16進制MACHINE CODE顯示。

〔E〕ERASE DISKETTE

這項功能容許用者把指定「磁軌」上的資料擦掉或全隻碟清除。由於這項功能能夠毀滅磁碟上的資料，所以螢光幕會打出THIS UTILITY DESTROYS DATA ON THE TARGET DISK 提示操作者。操作者在清洗之前，首先要輸入磁碟機編號、磁軌起點、磁軌終點及磁軌增量等資料。當進行時，LOCKSMITH 會以E 代表擦掉。

〔N〕NIBBLE EDITOR

這項功能在LOCKSMITH 用者來說又是一項有用的工具程式。對於一些想深入研究磁片的使用者而言，可以說是一個很有價值的工具。藉着NIBBLE EDITOR 的功能，使用者可以讀取某一磁軌的資料，然後進行檢查或修改。

在執行NIBBLE EDITOR 功能時，螢幕上將會有一對括弧，用以指出游標所在位置，且這個游標可以利用I、J、K、M及→和←鍵，來控制其移動方向及位置，以觀察各磁區的資料，若要使螢幕連續捲動，可按下〝，〞或〝・〞。若按下任何其他鍵則可中止螢幕捲動。

NIBBLE EDITOR 除了上述所提到的功能外，在螢幕上還可以看到〝(〞符號被用在1個NIBBLE的前面，以表示TRACK 的開始，而〝)〞則表示TRACK的結束。這些磁軌的起始和結束指示，也可以將游標移到使用者指定之位置，然後按下〝(〞或〝)〞來指示。

在NIBBLE EDITOR 中，另外提供了兩個按鍵功能：CTRL－B（把游標移到磁軌的起始處）；CTRL－E（把游標移到磁軌的結束位置）。

NIBBLE EDITOR 在螢幕上是以反相顯示來表示SE－LF－SYNC NIBBLES。若想將某一個NIBBLE改成SE－LF－SYNC NIBBLE，則可以將游標移到指定之NIBB－LE 上，然後按下S 鍵即可。另外，若要將1個SELF－SYNC NIBBLE 改成正常顯示，只要按下N鍵。

在NIBBLE EDITOR中，若按下CTRL－F 則可以進行字串搜尋的工作。要使用這個功能，首先將游標指向欲搜尋字串的第一個NIBBLE之上，然後按下CTRL－F則游標將會移到下一個字串處，假如沒有找到，則喇叭會發出聲音。這個功能也可以用來搜尋一個磁軌中的各個位址頭（ADRE－SS FIELD HEADER）。

在NIBBLE EDITOR 中還有兩個特別的鍵，分別是G和H。G的功用是把NIBBLE EDITOR 裏的字符化成文字顯示。而H的功用把高解像的圖形顯示出來。

上面已將NIBBLE EDITOR 的功能，作了一個整體介紹。最後要介紹NIBBLE EDITOR 中一個很重要的功能，即修改資料的能力。這項功能就是CHANGE MODE，要進入這個工作模式，只要將游標指向欲更改的NIBBLE位置且按下C鍵，則在螢幕狀態顯示上一個TRACK，在此工作形態下，若决定要更改所指定的資料則只要鍵入新值，然後按下RETURN鍵即可。

〔S〕DISK SPEED

這項功能能夠測試磁碟機的速度。磁碟機共有三種標準操作速度：一、正常操作速度一每分鐘三百轉；二、最佳操作速度一稍低於正常操作速度，但在複製磁碟時，適宜採用這個速度；三、與使用「母磁碟」的磁碟機具有相同的操作速度。跟着會選擇粗的（COARSE）、中間（MEDIUM）或細密的（FINE） 圖形刻度顯示，在上述選擇過後，即可指令LOCKSMITH 開始進行磁碟轉速測試，然後使用者即可利用螢幕上顯示的磁碟轉速特性曲線，調整磁碟機速度，調整完後，可按ESC離開。

〔C〕CERTIFY DISK

選擇這項功能，可以檢查磁碟或磁碟機的好壞，但是這項功能會毀滅磁碟上的資料，而在CERTIFY期間，〝C〞代表READ／WRITE 正確無誤，即表示你的磁碟及磁碟機非常良好。

〔I〕INSPECTOR／WATSON

這項功能執行時，一定要擁有INSPECTOR 磁片及16K RAM CARD 使用這項功能，將可使LOCKSMITH 的工作更具彈性化，使用INSPECTOR／WATSON，就如使用DISK ZAP一樣，只要多了些功能而已。

經過筆者一番解釋後，各位讀者應該對LOCKSMITH 5・0 有所認識，而LOCKSMITH由於市面的保護技巧千變萬化，LOCKSMITH 也相繼推出許多加強的版本，而最新版本是 LOCKSMITH 5・0 LEVEL X，但無論如何，LOCKSMITH始終是威力無窮的拷貝程式。■

增加DOS的CATALOG暫停功能

各位朋友相信在觀看磁碟目錄時都會因爲檔案的名字往上捲而感到十分煩人，若果能夠在CATALOG時按CTRL—C 鍵可以停止那可好多了。

本程式（CATALOG STOPPER)是令CTRL—C 鍵（或任何你自己選擇的鍵）在任何一部份CATALOG 時中止並給回控制權予使用者，而按入任何一鍵會使CATALOG功能繼續。

☆ 程式的使用方法很簡單，只需按入RUN CATALOG STOPPER即可，你可以將程式第20行改動來使用其它鍵代替CTRL—C ，例如將KEY＄＝CHR＄（13）（RETURN鍵）或KEY＄＝“”（SPACE鍵）。

☆ 閣下大可以將本程式加入你的HELLO 之內而當你BO—OT碟時即可獲得此項功能，不過本程式是設計給予標準DOS 之用的，主要是將＄AE63 改動，使DOS 跳往＄BA69檢查有否按入CTRL—C ，故此其他版本的DOS 可能會有不正常的現像出現。■

```
10  REM  ::::::::::::::::::
11  REM   CATALOG STOPPER
12  REM  ::::::::::::::::::
20 KEY$ =  CHR$ (3)
25 T = 256:DOS =  PEEK (978) * T +
     4410:AE = 174 - (157 -  PEEK
     (978)):L = DOS + 3119: POKE
     DOS,L -  INT (L / T) * T: POKE
     DOS + 1, INT (L / T)
30  POKE L + 0,32: POKE L + 1,12:
     POKE L + 2,253: POKE L + 3,
     201: POKE L + 4, ASC (KEY$) +
     128: POKE L + 5,208: POKE L +
     6,3: POKE L + 7,76: POKE L +
     8,44: POKE L + 9,AE: POKE L +
     10,76: POKE L + 11,60: POKE
     L + 12,AE: POKE L + 13,0
35 FOR I = 1 TO 40: PRINT "-";: NEXT
36 IF KEY$ < " " THEN  PRINT "CT
   RL-"; CHR$ ( ASC (KEY$) + 64
   );
37 IF KEY$ >  CHR$ (31) THEN  PRINT
    CHR$ (34);KEY$; CHR$ (34);
40 PRINT " WILL NOW MAKE A CLEAN
    BREAK IN": PRINT "A LONG CA
   TALOG. ANY OTHER KEYPRESS WI
   LL CONTINUE AS USUAL.": PRINT
   : PRINT "  NOTE: YOU CAN APP
   END LINES 20-30 TO    YOUR
   BOOT PROGRAMS TO INCORPORATE
    THIS  FEATURE INTO YOUR SYS
   TEM."
50 FOR I = 1 TO 40: PRINT "-";: NEXT
```

最快的COPY程式——DIVERSI COPY

近日市面出現一隻最新的COPY PROGRAM—DIVER—SI—COPY。它是由DIVERSI FIED SOFTWARE RES—EARCH INC出品，是目前最快的一個COPY程式。這家公司最爲人熟識的軟件有DIVERSI—DOS。

利用DIVERSI—COPY 抄錄一張沒有保護的磁碟（例如電腦時代程式磁碟），最快只需要33秒，它的使用較任何其他COPY程式都容易。DIVERSI—COPY自動檢查每一個BYTE，故此能知道抄錄過程是否沒有出錯。在抄錄的同時，它會顯示磁碟機摩打於每條軌轉動的速度快慢，減少了許多的問題。

當然，除了抄錄磁碟外，它還可同時用來FORMAT PRODOS，DOS 3·3，PASCAL或CP／M的DATA DISK 。每張只需18秒時間便可FORMAT好。

這個COPY 程式尚有一個特別之處就是能供大量生產用，如配合上128K RAM咭，COPY一張磁碟只要18至20秒，快過目前所有的同類型程式。程式唯一缺點是它不能COPY有保護的程式。

各位可參考表㈠便知道這COPY 程式的異常之處。在美國，這個程式售30美元。■

表㈠

	PRODOS COPY	DIVERSI COPY
一部磁碟機（磁碟半滿）	172秒 入18次碟	36秒 入 2次碟
一部磁碟機（磁碟全滿）	172秒 入18次碟	44秒 入 4次碟
二部磁碟機	80秒	33秒

ENTERTAINMENT

最新遊戲軟件評價　　李德華

七海尋金熱

SEVEN CITIES OF GOLD

SEVEN CITIES OF GOLD是一個ACTION ADVERTURE GAME。開始時，你要選擇PLAY A GAME OR CREATE A WORLD（通常選擇前者），如玩者選擇後者，電腦會從磁碟讀取一些資料，跟着叫你放入空白磁碟，然後按RETURN，玩者大約等十分鐘後，電腦會命令玩者重新啓動SEVEN CITIES OF GOLD。

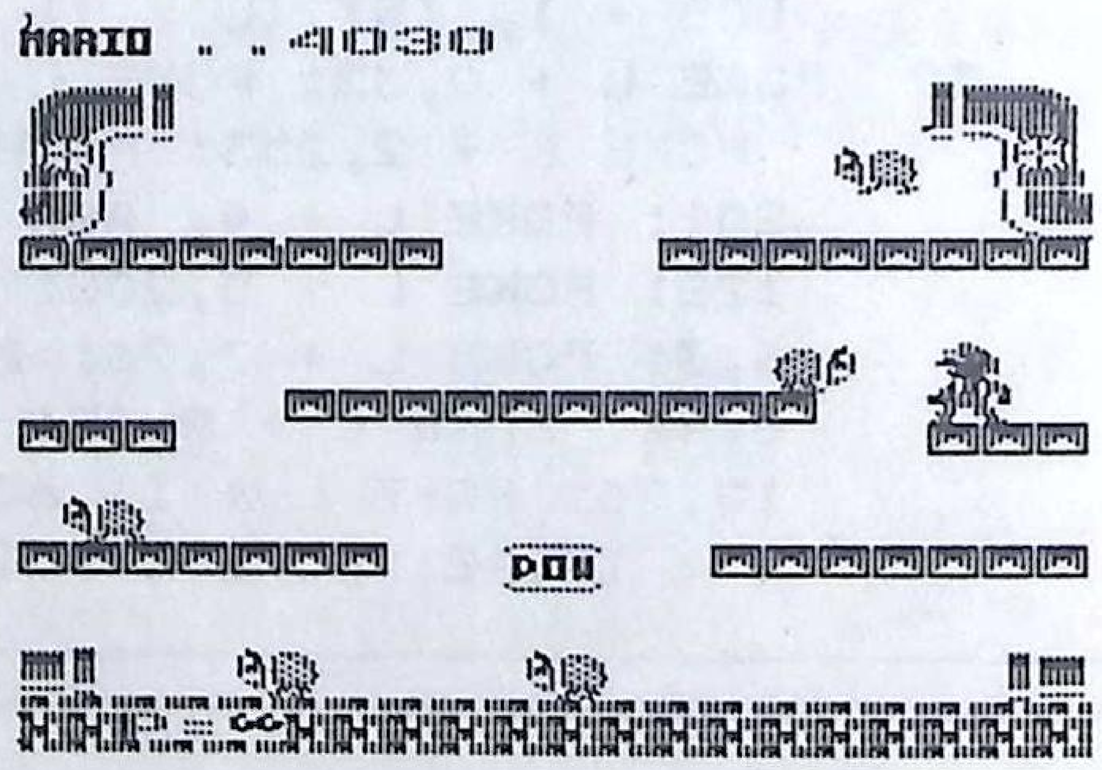
而電腦會重新問同樣的問題，當回答完後，放入剛製好的MAP DISK（通常MAP DISK會連同遊戲一同售出）。而這個遊戲共提供了4種LEVEL，MASTER是最易玩的。當所有步驟完成後，電腦會從磁碟再讀取資料，此時你會在一條村莊裏，而你用JOYSTICK控制船長，BUTTON 0或BUTTON／加快行路的速度，而在你現在的路途中，會分別看到PUB、HOME、OUTFIT 3間屋。PUB是酒館，對於你沒有多大作用，HOME是家，這裏用作繪畫地圖，稍後再詳盡說。在這3間屋之中，最重要是OUTFIT，這間屋可以說是商店，這裏可以用你的2000個金幣買東西，包括船、人、食物、貨物。如果你喜歡電腦來定的數目，可以離開這間商店，跟着你便上船開始你尋金及去找尋新大陸的夢想。

待一會後，你便會在海上航行，你應該向西邊行駛，而螢幕周圍分別顯示你的概況，而最重要一項是食物，頂上的文字顯示時間，而下方的文字顯示水位概況，當水位顯示MEDIUM的時候，則表示附近有陸地。當看到陸地後，你便預備登陸的工作，此時電腦應該顯示3行文字，選擇VIEW MAP來觀察你所發現的新大陸，跟着選擇DROP STUFF OFF來分配登陸所需的設備，用JOYSTICK分別選擇人、食物及用具的分配，當所有步驟完成後，使用JOYSTICK移動一個像黑圈的物體登陸，如果陸地非常大則會出現有村落，甚至有金礦，如果入了村落，一些土人會對你進行襲擊或友善對待，如不幸是前者，用BUTTON／（武器）攻擊他們，或到一些像白色棒狀物體，按BUTTON 0離開，如是後者，你可以找他們的族長進行貿易，首先給他們禮物，表示沒有惡意，跟着再和他們進行貿易。當貿易完後，應盡快離開，因你的食物及人此時應耗盡了。上了船後，應把剛才所有的裝備收回船上，跟着以最快速度回程，但海上可能會行雷閃電，此時應設法離開這海面。如果你航線正確的話，電腦會給予你指示接近西班牙。

當回到陸地後，立刻回家，電腦便把你剛才所發現的土地及所有的環境繪畫出來，跟着你可以到PALACE領取獎金，因這個使命是西班牙的皇帝叫你去做的。再去PUB SAVE GAME，跟着你再上船，重複同樣過程，直至把所有新大陸（七大洲）及金礦找出來。

玩這遊戲的秘訣是應盡量買多些食物，並祇有一些較大的洲才上岸，記着所行的路線，而熟悉地理的人士，玩這個更易發揮你的探險精神，筆者在這裏祝你會一帆風順！

遊戲形式　ADVENTURE GAME
三粒星　＊＊＊ ■



頂烏龜　　李德華

遊戲名稱：MARIO BROS
遊戲形式：殲滅式娛樂性遊戲
評分：＊＊＊

MARIO BROS是一個由「街機」改編而成的遊戲，而遊戲最大的特色是能夠容許雙人同時進行遊戲，這令到MARIO BROS更富娛樂性。

控制方面包括：

1或2　選擇1人或2人進行遊戲
JORK　選擇控制桿或鍵盤控制
CTRL—O　回到選擇畫面
CTRL—Q　重新開始遊戲
CTRL—S　聲的開關
←及→　選擇LEVEL

遊戲的目的是要把烏龜、蟹、蜘蛛消滅，首先用頭搥令它們翻轉，跟着用脚踢它們落水便行，遊戲進行時，會分別有火球和冰怪襲擊你（冰怪在第十六難關才有），你只要盡量避開它們或用頭搥對付它們。在遊戲進行時，更有錢從天而降，當吃到錢後，便會有800分的獎勵。

在LEVEL 16開始，更有錐形物體襲擊你，萬一非常多怪物或火球之類東西，而你又不能對付，玩者可以用POWER.來消滅它們，但POWER只可用三次。如是者把所有怪物消滅後，便能夠升LEVEL。

總括來說，這個遊戲的音响和圖像和街機簡直可以比美，而筆者發現這個程式沒有任何保護，所以玩者可以輕易COPY過來。■

古墓之謎

蟻皇

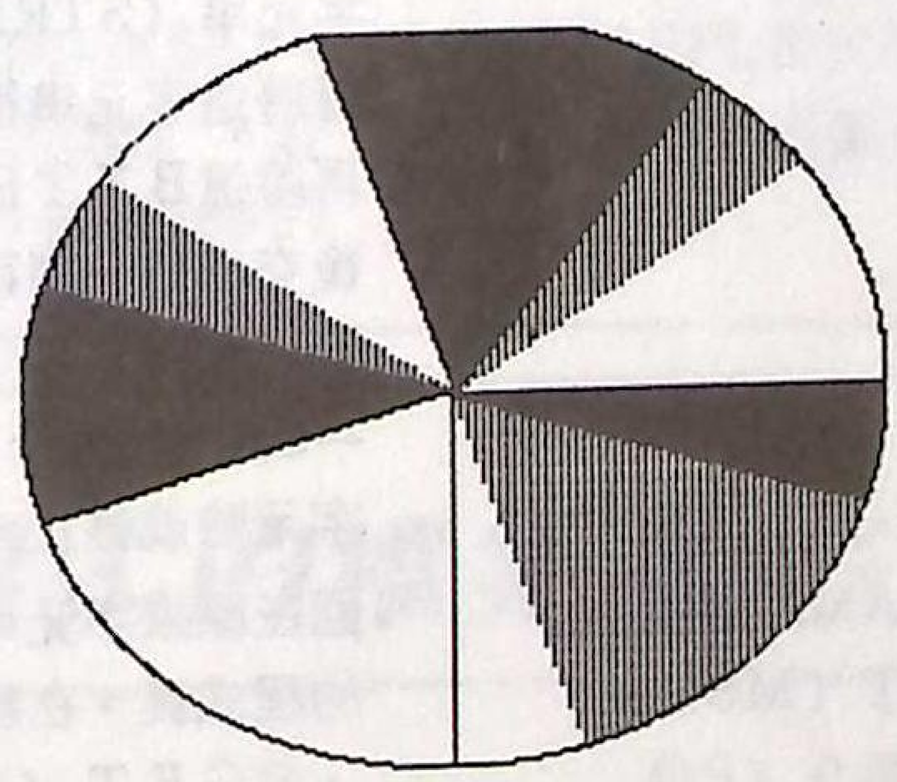
今次爲各位介紹的GAME 叫LOST TOMB，這雖然不是很新的GAME ，但也是84年的作品。這GAME 的中文名叫「古墓之謎」。譯名十分有趣，閒話休提。

當放入碟後約4.5秒，便會有一個大〔金字塔〕出現。遊戲可以一人或兩人玩。之後螢幕便打出一個人（即你）跌落一個金字塔，如不想看，可按BUT—TON（O）；稍後，又打出一大堆文字，叫你取有鎖匙的桶等等……。遊戲開始時，你會看見自己在ROOM1 ，你有5個炸彈，和數不盡的子彈。CONT—ROL BUTTON：B（O）用來FIRE，B（1）用來投炸彈（螢幕最下行有顯示）。你的目的是去取鎖匙，但萬萬不能觸到蠍子或蜘蛛。還有一些〝扁型〞的陷阱，你一觸到，金字塔便會發震。那時有BONUS X2，X3……看時間。有時間限制的，左邊的TIME就是，右邊是子彈，不要理它。圖中有5個桶，內有KEY，子彈或炸彈，當蠍子或蜘蛛襲擊你時，便用子彈射牠，必要時才用炸彈。在ROOM1，2，3盡量不要用炸彈，ROOM4，5，6…才用，因那時突然有蝙蝠飛來襲擊你。取完最頂的四個桶才取最下一個，取完後便行到最低的中心（即第五個桶下面）。好了，過了第一關，你要有心理準備，遊戲會轉入隧道內進行（到ROOM2，3的隧道）。你不可再入ROOM1 ，要落樓梯去到ROOM2，3。這時會有無數的蝙蝠撲向你，閣下要盡快去ROOM2，3 ，在那裏才好用炸彈。

這遊戲每個ROOM 的版圖都不同，十分有新鮮感。整個金字塔有數十個ROOM ，筆者只差十多個ROOM 便行完全個金字塔了，眞慘。這個遊戲的缺點就是每死一個人，便要從頭打一大堆字，不過放心，不會從ROOM1 開始呢！

秘訣：玩這遊戲時，最緊要好似玩「BRUCE LEE」，不要貪心取分射蠍子和蜘蛛。速戰速決吧！■

餅形圖



莫小覷這家短短的程式，它所製作的餅形圖是所見過的同類程式中最好及最短的一家。

整個餅形圖最高可分15份，每份會自動設定顏色，方便觀看。最細的一份可以是0．5％，使用時只要輸入有多少份及每份所佔的比例即可。

程式使用高解像圖頁，各位不妨研究程式的每行結構，學習其中所採用的技巧。

行2是指令每份所用的顏色。

行6設定餅形圖的圓周共分爲1000點，每份所佔的比例。

行7繪畫圖形。

行8 檢查每份圖形並將整個部份變成所設定的顏色。■

```
0   HOME
1   DIM N(15),N%(15),P(15),CO(15)
2   FOR J = 1 TO 15: READ CO(J): NEXT
     : DATA 2,3,0,2,3,0,2,3,0,2,3
     ,0,2,3,0
3 H = 140:K = 96
4   INPUT "HOW MANY NUMBER TO CHAR
     T?(MAX=15)";N
5   FOR M = 1 TO N: INPUT "ENTER";
     N(M):TM = TM + N(M): NEXT
6   FOR M = 1 TO N:N%(M) = N(M) /
     TM * 1000 + .5:PT = PT + N%(
     M):P(M) = PT: NEXT : HGR2
7   FOR J = 1 TO 1000
8   FOR T = 1 TO 15:F = F + (J = P
     (T)): NEXT
9 X =  SIN (J / 159) * 113
10 Y =  COS (J / 159) * 96
11  HCOLOR= CO(F): HPLOT X + H,Y +
     K TO H,K
12  HCOLOR= 3: HPLOT X + H,Y + K
13  NEXT
14  REM ::::::::::::
15  REM    SIMPLEST
16  REM   PIE  CHART
17  REM ::::::::::::
```

蘋果裡的「蟲」

水平客

在APPLE Ⅱ PLUS電腦裏是蘊藏着一些錯誤，這在一個複雜的系統裏是很難完全避免的。在未指出這些錯誤之前，讓我們先看看在MANUAL裏的一些錯漏。

使用手册上的錯漏

在APPLE Ⅱ REFERENCE MANUAL 裏第7頁的表格中，那個→鍵和←鍵的美國資訊交換標準碼（ASCII CODE）是換轉了，→鍵的ASCII CODE應該是＄95，而←鍵的ASCII CODE是＄88才對。另外在第36頁中第10行上寫着"IT FLIPS ANNUNCIATORS 0 AND 1 ON AND ANNUNCIATORS 2 AND 3 OFF ON THE GAME I／O CONNECTOR…"，這句中的"ON"和"OFF"應該互換才算正確，與事實相符。也就是說在那個遊戲輸入／輸出連接器上（可參看MANUAL裏第100頁的圖16），當電腦的電源剛接通時，輸出點（接脚）AN0和AN1是在OFF（約0.3 VOLTS)狀態，而AN2和AN3則是在ON（約3.5 VOLTS）的狀態。

在BASIC PROGRAMMING REFERENCE MANUAL裏第135頁所列出POKE 指令的結果都是錯誤的，現在只將正確的資料列出如下：

POKE—16296，1 轉AN0到0．3VOLT
POKE—16295，0 轉AN0到3．5VOLT
POKE—16294，1 轉AN1到0．3VOLT
POKE—16293，0 轉AN1到3．5VOLT
POKE—16292，1 轉AN2到0．3VOLT
POKE—16291，0 轉AN2到3．5VOLT
POKE—16290，1 轉AN3到0．3VOLT
POKE—16289，0 轉AN3到3．5VOLT

另外在第137頁裏所說STRING VARIABLE NAME 是用2個BYTE 來代表，第一個BYTE的第一BIT（MSB）是1（NEG） 其實是錯，這個MSB BIT 應該是0（POS），而第二個BYTE的MSB BIT則應該是1（NEG）才對。

在此順便一提MANUAL裏沒有提及的一些較重要資料：

＊ APPLESOFT 的一句中是可以擁有幾個指令，但並不是任何指令都可以和其它指令同處在一句中。例如REM指令後面是不能有其它指令。另外ONERR GOTO 指令後面亦不能有其它指令。試看下面句子：

10 ONERR GOTO 200：GOTO 90

在這句中的GOTO 90 指令是永不會執行的。

＊ 最好避免用記憶體第零頁的第214（HEX是＄D6）BYTE。因爲當這個BYTE 的值大於127時，只要你按入任何指令，寄存在記憶體的BASIC程式就會自動RUN 起來。當然，這也有它的好處，就是當你不想別人偷看或抄襲你的程式時，你先LOAD入你的BASIC程式，然後按入POKE 214，255 後就可以了。自此以後，按入任何指令如LIST，POKE，SAVE…等都只能使你的程式RUN起來，而不能有其他作爲了。

＊ 在讀取GAME PADDLE的數值時，不要在太短的時間內連續讀取兩個或以上PADDLE 的數值，因爲這樣會使你所讀得的資料不準確。這在編寫組合語言時尤要注意。

主機中的毛蟲

現在讓我們來看看APPLE主機裏的「蟲」：

＊ 若你的APPLE是48K而沒有DOS，則下面的程式會產生奇怪的結果：

```
10 GET A$
20 PRINT VAL (A$)
```

若你按"1"字，電腦打出的却是1．11111111E＋16。若你按"2"字，電腦打出的是2．22222222E＋16。若你按"9"字，電腦打出的是1E＋17。這個錯誤可以說是間接由APPLESOFT 的VAL指令所產生的。我們先看看VAL指令本身內裏的工作過程：首先它找出字元串（STRING) 在記憶體的所在地及字元串的長度，跟着將這字元串後面的一個BYTE的值寄存在STACK裏，再將這BYTE改爲零（目的是表示這字元串的終止），然後交由另一副程式將這字元串化爲浮點數。這個副程式便從字元串的開始，續一將每個字元（註：這些字元都是用ASCII CODE 代表的）化爲數值，直至遇到一個非數字的字元時就停止。最後，VAL會將寄存在STACK裏的值放回在原來字元串後面的那個BYTE裏。現在我們回到原來的程式裏。在程式開始時，HIMEM是指着地址＄BFFF 的，當GET（A＄）取得A＄字元時，就將這A＄記存在＄BFFF裏。而在這BYTE的後面是＄C000 ，但這裏是沒有記憶體的。事實上，＄C000是APPLE鍵盤的輸入暫存器（INPUT REGISTER），而＄C001到＄C00F 只不過是＄C000 的重叠，所以它們的值都是相同的。當VAL 想把零值寄存在＄C000 上時是不可能的，＄C000 的值只能被鍵盤所改變。假若GET（A＄）所取得的是"1"的字元，那麽由＄C000至＄C00F就都是保留着"1"的字元了，所以當VAL在化字元串爲浮點數時，誤把由＄C000至＄C00F的都當作是A＄的一部份，直至到在＄C010 處時遇到非數字的字元才終止。而結果所得的是個17數位的字元串，所以電腦就打出1．11111111E＋16作爲A＄的值。假若原來的A＄是"9"的話，電腦就作四捨五入，打出1E＋17了。要解决這個問題有幾個方法：

蘋果裡的「蟲」

第一個方法是在原程式裏加入一句

15 A$＝A$

這句是令APPLESOFT 多製造一個A＄而記存在＄BFFE裏，而當VAL 把零值寄存在A＄後面（即＄BFFF)時，是沒有問題的，因爲＄BFFF是個實在的記憶體。

第二個方法是用HIMEM：49151 來代替原來的49152（即是以＄BFFE來作爲RAM的終點）。

第三個方法是用磁碟機和DOS，當DOS LOAD入記憶體後，HIMEM 就會指向＄9600了。

＊ APPLE 的第二個「蟲」是在很特殊的情況下才會產生。當你將一個浮點數放入一個整變數裏，然後再PRINT 出這整變數的值時，就會有捨去（TRUNCATE）的結果產生。試看下面句子：

10 A%＝—32768・00049
20 PRINT A%

正確的答案應該是—32769，但電腦打出的却是32767。這個錯誤是APPLESOFT 在運算這個整數的最大負值作捨去時的錯漏。還好這個錯誤只有在下面的狹窄範圍內才會產生:

—32768・000493 ＜浮點數
＜—32768・000484

＊ APPLE 的第三個「蟲」其實在其他微電腦裏也可能存在着，只要它們是採用6502 微處理器作爲電腦的CPU的話。也就是說這個「蟲」是在6502微處理器本身。錯誤是發生在機械語言裏的〝JMP INDIRECT〞指令上。讓我們先看看這〝JMP INDIRECT〞的指令：

JMP （＄XXYY)

這裏XX是代表高階地址（HIGH-ORDER ADDRESS)，而YY是代表低階地址（LOW-ORDER ADDRESS）。若YY不等如FF的話，那麽一切都正常。例如，根據以下的資料：

地址	內容
4000	22
40CC	55
40CD	66

執行JMP （＄40CC）指令的結果是PC←＄6655，（PC是程式位址計數器—PROGRAM COUNTER）。它的操作過程是將＄40CC地址上的BYTE作爲PC的低階地址，然後又將（＄40CC＋1）即＄40CD地址上的BYTE作爲PC的高階地址，也就是程式將會在＄6655地址上繼續執行。但是當YY等如FF的時候，問題就產生了。例如，根據以下的資料：

地址	內容
4000	22
40FF	55
4100	66

執行JMP （＄40FF）指令的結果却是PC←＄2255，而不是PC←6655。錯誤是在找尋PC的高階地址時所產生的。本來在計算（＄40FF＋1）的結果應該是＄4100，但因爲沒有把進位計算在內，結果所得的是＄4000，所以把4000地址上的BYTE作爲PC的高階地址了。

要解決這個問題，就要在編寫組合語言時留意〝JMP INDIRECT〞指令裏的間接低階地址不要等如＄FF。但若是這個間接地址會在程式執行時有所改變的話（即自動修正程式），則錯誤就很難避免了。

以上是筆者所搜集到的資料，相信讀者亦可能會發現到APPLE其他的「蟲」，希望讀者能將之供諸於世，使到大家都有所警覺，不致陷入迷惑，避免浪費無謂的時間。■

改動BOOTING程式的檔案名字

莫信

一張磁碟，當 INIT時，它的BOOT PROGRAM 便已被決定了，如用 HELLO 來作 BOOTING PROGRAM，我們便要 INIT HELLO ，若 HELLO 被 DELETE 或RENAME ，則 BOOT 碟時便會出現 ERROR。

每張磁碟必須有一個 BOOTING PROGRAM，而 INIT磁碟時，它的 FILENAME 便被放置於 TRACK $1 的 SECTOR $9 來指示 DOS 當磁碟 BOOT 起後便執行這個 FILENAME 的程式，所以當 BOOTING PROGRAM 被 DELETE 或RE G 了，DOS 便找不到這個FILENAME，正常操作受到影响，出現錯誤，這個BOOTING PROGRAM的FILENAME 正確被存放的位置是在 TRACK $1 SECTOR $9 由位址＄77 開始，如圖（它的 BOOTING PROGRAM是HELLO），我們只要利用 COPY PROGRAM 來改動這些位址便成了（長度是 30個位址，因一個 FILE 的 FILENAME 最長是由 30個 CHARACTER 組合而成的）。

```
70- 00 00 00 20 00 C8 C5 CC .....HEL
78- CC CF A0 A0 A0 A0 A0 A0 LO
80- A0 A0 A0 A0 A0 A0 A0 A0
88- A0 A0 A0 A0 A0 A0 A0 A0
```

APPLESOFT與MBASIC之比較

擁有Z-80 CARD 的用家都知道CP／M SYSTEM MASTER 內有一MBASIC 語言運用程式。大家應該知道APPLESOFT BASIC乃是由MBASIC 改良而成，筆者嘗試爲二者作一粗略比較。文章只是大致敍述一些MBASIC有而APPLESOFT BASIC 沒有的COMMAND。

㈠事實上，MBASIC幾乎擁有DOS TOOL KIT上的LOADAPA 之功能，如AUTO LINE NUMEMBER，MERGE，RENUMBER等之有效功能，用法之分別是DOS TOOL KIT 要用&爲第一字（卽& AUTO…），而MBA—SIC則可將其當爲普通一般的COMMAND用（卽AUTO，MERGE…）。

㈡MBASIC具有SWAP的功能。SWAP 的功能主要是將兩變數裏的值互調。

```
10 A = 10:B = 50
20 SWAP A,B
30  PRINT A,B
RUN ——→ 2,1
```

㈢MBASIC中，LINE INPUT 這個指令，功能相似於APPLESOFT 指令之INPUT 。惟一不同之處是這LINE INPUT 可隨意打入任何所需要的字符，包括逗號。而整個字串都會保留在指定之字串變數裏。用法格式爲LINE INP—UT A$……

㈣MBASIC中有一個BEEP 的指令，功能類似CTRL—G，但這個COMMAND就方便得多了！用法可參考下列程式。

```
10  FOR A = 1 TO 5
20 BEEP A,10
30  NEXT I
40  END
```

其中LINE 20之10爲音量之DURATION。

㈤在MBASIC 下，對於程式中有不正常之情況出現時，可有ON ERROR GOTO，RESUME 敍述與ERL、ERR等保留字可供使用，但相對之，APPLESOFT BASIC只有ONERR GOTO與RESUME敍述可供使用。

MBASIC是利用ERL（ERROR LINE NUMBER)獲知目前發生錯誤狀況的列號，卽ERL及利用ERR 獲知錯誤狀況的編號。錯誤情況譬如除法運算時除到0時等。大家可參考下列程式：

```
10 ONERROR GOTO 50
20 INPUT A
30 ?1/A
40 END
50 IF ERR=11 AND ERL=30
   THEN?"'A'MUST NOT BE ZERO"
```

上列程式中，LINE 50：當ERL=30 且ERR=11 成立時，表示發生之錯誤列號爲30，而錯誤編號爲11（卽DATA DIVISION BY ZERO）。

㈥MBASIC 中有一IF…THEN…ELSE 之敍述。這種若…則…否的形式在MBASIC或PASCAL可以直接寫出來，但在APPLESOFT BASIC中則不可。其用法格式如下——若A＞B 則L＝A 否則L＝B （卽IF A>B THEN L=A ELSE L=B）

㈦MBASIC中有一個頗特別的指令——PRINT…USING。用法及格式請參考下列程式。

```
10 READ A
20 PRINT A USING " #####,###"
30 DATA 1000.55,110,1.2
RUN ——→   1000.500
           110.000
             1.200
```

這個COMMAND之功能在APPLEBASIC中是需要用一個PROGRAM才能達成的。

㈧在APPLESOFT BASIC中如要將GR情況下之底色轉變的話，是非常之麻煩的。但在MBASIC中，你只要鍵入GR 1，7，則電腦會將底色轉爲COLOR 7。

㈨在MBASIC之DOS 系統下，有幾點與APPLE DOS不同的。首先MBASIC DOS下無論用SAVE／BSAVE／LOAD／BLOAD／RUN／BRUN，都需要用引號括起FILE NAME。例如：SAVE "ABC"

接着，MBASIC 有一COMMAND 可令PROGRAM受保護，只可RUN，不可LIST —— P。用法：SAVE "ABC"，P

至於DELETE及RENAME程式也與APPLESOFT 不同。

DELETE：MBASIC F，用KILL取代DELETE，但MBASIC要指出程式是BASIC或MACHINE CODE。用法如下：KILL "ABC.BAS"。（BAS代表BASK，非FI—LE NAME）

RENAME：MBASIC F，RENAME的功用可用下列方法達成——NAME "ABC" AS "XXX"

㈩而最後要讓的是有關列印機的。例如我們用LLIST則表示LIST在PRINTER上，又LPRINT則PRINT在PRIN—TER上。

好了，討論到此爲止，雖然並不能將所有不同處列出，但大致上而將普遍的不同處寫出。

張峯寧

Software Reviews

教育眼

預備到美國升學的同學適用的兩套軟件

• SAT WORD ATTACK SKILLS
• ENGLISH SAT ♯1 & ♯2

近年來，出外留學的學生日漸增多。其目的地大多爲美國及加拿大。而有意往美國留學的讀者相信都聽過一個叫—SCHOLASTIC APTITUDE TEST（簡稱SAT的考試）。故今次爲大家介紹兩套用作預備參加SAT試之教育軟件。

第一套名爲SAT WORD ATTACK SKILLS，分A，B兩面，出版公司爲〝EDU〞，A SIDE爲SAT PREFLEXES，B SIDE爲SAT ROOTS。此軟件是專用作預備SAT EN—GLISH中VACABULARY的，每面分7個TOPIES，每一個TOPIES分別介紹一個PREFLEX或ROOT，而每個的TOPIES有十個有同樣PREFLEX或ROOT的字，即兩面共介紹140個字。

教導方法是以測驗形式，首先當用家選定TOPIES後，則電腦會先解釋這個PREFLEX或ROOT之解釋，接着會問你想做同義詞或相反詞之練習，而練習（測驗）形式分兩種，一爲TEST，一爲INSTRUCTION。INSTRUCTION測驗是每當你做完一題（選擇題），電腦會即刻告知錯或對。如果錯，則會舉一例，然後再叫你選另一答案，之後並會解釋一字之意思。如果答對，則電腦會問你是否需要再做一題有另外五個不同選擇的問題。而TEST，則當你做完之後，給予答對之百分比，並不作任何解釋，此套軟件是用APPLE PILOT編寫的。

而另外一套則名爲ENGLISH SAT ♯1& ♯2，也是分兩面，出版社是THE MICRO LAB LEARNING CENTER（MICROLAB）。此套軟件與上述一套最大之分別爲——前者只是學VOCABULARY，而後者則包括ANALOGIES，ANTONYMS，COMPREHENSION，ENGLISH GRAM—MER，COMPLETIONS，換言之，幾乎包括了SAT EN—GLISH所要考的形式。而其教導方式也是以測驗方式，當你選定COURSE 後，便開始。每當你選擇答案後，電腦會立即告知對或錯，如錯，則會解釋你所選的答案，然後叫你再答，有四次機會，即選到剩最後一個。但如果你答對，電腦也會解釋你的答案，然後繼續第二題。

總而言之，本人覺得此兩套軟件是有意參加SAT試的讀者必備的。但兩者比較，後者之測驗範圍較大，但畫面則非常單調，不及前者。■

捉蟲訓練—PYTHAGORAS

今期捉蟲問題是考驗你對蘋果電腦處理變數的常識。以下是一條計算PYT—HAGORAS 定理的程式。一個直角三角形，只要知道其中兩條邊，便可輕易的求出第三條，但爲何這程式不能呢？問題出現在那裏？

```
20  TEXT : HOME
30  INPUT "ENTER ONE SIDE OF A RI
    GHT TRIANGLE ";LEG1
40  INPUT "ENTER THE OTHER SIDE "
    ;LEG2
50  HYPOTENUSE =  SQR (LEG1 ^ 2 +
    LEG2 ^ 2)
60  PRINT : PRINT "THE HYPOTENUSE
    IS";HYPOTENUSE
79  END
```

我們歡迎讀者提供〔捉蟲訓練〕問題。程式長度沒有限制，以越簡短越好。程式必須要以APPLESOFT 寫成，並符合下列各項要求：

1. 程式必須是APPLESOFT ，任何POKE或CALL機械語言的敍述必須正確。
2. 程式必能出現某些結果，而不可出現SYNTAX ERROR。
3. 所有問題必須附加簡短說明及答案。

假如閣下的作品獲得刊登，我們將會送出十二期免費雜誌給你或將你的訂閱延長一年。

所有投稿請寄九龍中央郵政信箱C. P. O BOX 71193 。信封面請寫上〔投稿捉蟲〕。■

答案：APPLE 處理任何變數，無論該變數多長，只會取頭兩個字母來作分辨，大家只要將LEG1及LEG2改爲L1及L2便可。■

COMPUTING AGE

COMPUTING AGE C.P.O BOX 71193 3-312007

設計信紙頭

本程式可以令你很方便的設計自己的信紙版頭，簡單而實用。適合有GRAFTRAX 咭和MX-80或相容系列的印字機。

行號1150-1210，1240-1280 及1590 是供大家輸入自己的資料。

```
1  REM   *************************
4  REM   *    LETTER HEAD        *
6  REM   *  BY COMPUTING AGE     *
8  REM   *************************
25 GOTO 1000
1000  HOME : REM  CLEAR SCREEN
1005  PRINT : PRINT : PRINT : PRINT
1006  PRINT : PRINT : PRINT : PRINT
 : PRINT
1010  PRINT  TAB( 5)"************
***************"
1020  PRINT  TAB( 5)"*
              *"
1030  PRINT  TAB( 5)"*";
1040  INVERSE : PRINT  TAB( 8)"FA
NTASTIC LETTERHEADS ";
1050  NORMAL : PRINT "*"
1060  PRINT  TAB( 5)"*
              *"
1070  PRINT  TAB( 5)"* BE WHAT YO
U WANT TO BE *"
1080  PRINT  TAB( 5)"*
              *"
1090  PRINT  TAB( 5)"************
***************"
1100  FOR PAUSE = 1 TO 2000: NEXT
 PAUSE
1110  PRINT : PRINT : PRINT : PRINT
 :
1120  PRINT  TAB( 5)"PREPARE YOUR
 PRINTER FOR LETTERHEAD"
1130  PRINT  TAB( 5)"TURN THE PRI
NTER OFF AND THEN ON"
1140  PRINT : PRINT  TAB( 5)"PRES
S THE <#> KEY"
1145  PRINT  TAB( 5)"TO COMPUTING
 AGE"
1150  PRINT  TAB( 8)"<1> COMPUSOF
T"
1160  PRINT  TAB( 8)"<2> COMPUTIN
G AGE"
1170  PRINT  TAB( 8)"<3> COMPUSOF
T"
1180  PRINT  TAB( 8)"<4> COMPUTIN
G AGE"
1190  PRINT  TAB( 8)"<5> DO IT YO
URSELF"
1210  PRINT  TAB( 8)"<6> EXIT TO
BASIC"
1220  PRINT : PRINT : PRINT  TAB(
5)"ENTER A NUMBER FROM 1-6"
1230  GET Y$:A =  VAL (Y$): PRINT
1235  HOME
1240  IF A < 1 OR A > 6 THEN  PRINT
 "1 TO 6 PLEASE!": GOTO 1140
1250  IF A = 1 THEN L$ = "COMPUTI
NG AGE": GOTO 1430
1260  IF A = 2 THEN L$ = "COMPUSO
FT": GOTO 1430
1270  IF A = 3 THEN L$ = "COMPUTI
NG AGE": GOTO 1430
1280  IF A = 4 THEN L$ = "COMPUSO
FT": GOTO 1430
1300  IF A = 5 THEN  GOTO 1400
1335  END
1400  PRINT : PRINT : PRINT  TAB(
5)"TYPE IN THE NAME YOU WANT
 TO USE"
1410  PRINT  TAB( 3)"TO A MAXIMUM
 OF 40 LETTERS AND SPACES"
1420  VTAB (20): INPUT "";L$
1430  HOME
1440  IF  LEN (L$) > 40 THEN  PRINT
 "TOO MANY CHARCTERS!": PRINT
 : GOTO 1400
1500  PR# 1
1510  PRINT  CHR$ (27) +  CHR$ (7
5)
1520  FOR X = 1 TO 75
1530  PRINT  CHR$ (27) +  CHR$ (6
)"150"
1540  NEXT X
1550  PRINT  CHR$ (13);
1560  PRINT  CHR$ (27) +  CHR$ (7
1) +  CHR$ (27) +  CHR$ (14)
;L$;
1570  PRINT  CHR$ (20); CHR$ (27)
 +  CHR$ (83) +  CHR$ (1)
1590  PRINT "COMPUTING AGE C.P.O
BOX 71193 3-312007"
1600  PRINT  CHR$ (27) +  CHR$ (7
5)
1610  FOR X = 1 TO 75
1620  PRINT  CHR$ (27) +  CHR$ (6
9)"150"
1630  NEXT X
1640  PRINT  CHR$ (13)
1650  PRINT  CHR$ (27) +  CHR$ (7
2);
1660  PR# 0
1665  HOME
1670  GOTO 1140
```

徵求一行程式有獎遊戲

在上月我們收到許多〔一行程式〕的投稿，其中不乏精彩的作品。

這個遊戲旨在鼓勵大家動腦筋，利用最簡短的程式，設計一些有創意的效果。

今次未刊登你的作品的讀者請勿灰心，可能我們會在下一期登出。

各位讀者寫來程式時，必須用多行的形式寫來（參考本文的各位得獎例子），而且要書寫清楚。投稿請寄來CPO BOX 71193 。入選者可獲得任何一期的程式磁碟一張（請註明要的磁碟）。■

葉柏青讀者寫了一個金從天降的遊戲，希望大家玩得開心。當RUN 之後，螢幕會顯示許多數字，請按2（如果不按便ERROR)，然後開始遊戲。電腦會拋下一串一串的大銀，玩者要接住大銀，每接住一次可得一分。控制移動的鍵是1（左）及3（右）。遊戲的速度是越來越快。葉柏青可取得第十期程式磁碟。

```
3     FOR W = 1 TO 50
  : PRINT PEEK (49152)
  : NEXT
  : X = 20
  : FOR C = 1 TO 99
  : HOME
  : VTAB 9
  : PRINT "SCORE:";S
  : A = INT ( RND (1) * 20) + 10
  : FOR B = 4 TO 22
  : HTAB A
  : VTAB B
  : PRINT "_"
  : FOR Z = 1 TO 99 - S
  : NEXT
  : HTAB X
  : VTAB 23
  : PRINT " "
  : X = X + PEEK (49152) - 178
  : HTAB X
  : VTAB 23
  : PRINT "="
  : NEXT
  : IF A = X THEN S = S + 1
  : NEXT
```

ANDREW LUI君寫來的是一個觀看TEXT FILE 的短程式。首先打入TEXT FILE 的名字，然後打入該檔案的長度，便可看到反白顯示的內容。ANDREW 可得到一張第七期的磁碟。

```
10    HOME
  : INPUT "FILENAME?";A$
  : INPUT "LENGTH?";L
  : D$ = CHR$ (4)
  : PRINT D$;"OPEN ";A$
  : PRINT D$;"READ ";A$
  : PRINT
  : FOR I = 1 TO L
  : INPUT O$
  : INVERSE
  : PRINT O$
  : NORMAL
  : NEXT I
  : PRINT D$;"CLOSE ";A$
  : PRINT "END OF FILE"
```

楊耀宗所投稿的程式是一個繪畫立體圖形圖案，可贏到第十一期的磁碟一張。

```
3     FOR T = 1 TO 6
  : READ A
  : POKE 767 + T,A
  : NEXT
  : HGR2
  : POKE 232,0
  : POKE 233,3
  : ROT= INT ( RND (1) * 9 + 1)
  : SCALE= INT ( RND (1) * 20 + 1)
  : X = RND (1) * 130 + 1
  : Y = RND (1) * 80 + 1
  : FOR T = 0 TO 6.3 STEP .03
  : DRAW 1 AT SIN (T) * X + 140,
     COS (T) * Y + 96
  : NEXT
  : RUN
  : DATA 1,0,4,0,4,0
```

陳立新所作的一個繪畫類似太陽的圖案為他贏得第4期的磁碟

```
3     FOR A = 7676 TO 7690
  : READ D
  : POKE A,D
  : NEXT
  : DATA 2,0,6,0,9,0,53,39,0,58,36,45
     ,45,7,0
  : POKE 232,252
  : POKE 233,29
  : HGR
  : SCALE= 25
  : FOR T = 1 TO 193
  : ROT= T
  : XDRAW 2 AT 140,80
  : NEXT T
```



解解拆拆

MURDER ON THE ZINDERNEUF

需要設備：

A） APPLE][+及磁碟機

B） BIT—COPY PROGRAM如LOCKSMITH 4.1 及 LOCKSMITH 5.0

C） 空碟一張

先用BIT—COPY PROGRAM把"MURDER ON THE ZINDERNEUF" 抄到空碟上（由TRACK $00—$04 再由$07—$22）。最後用LOCKSMITH 5.0 把它由TRACK $5.25—TRACK $6.25 抄到碟上。

現在就大功告成了！■

更正

第十一期P.44，右邊欄中間的列表由底數上第四行

STA $20 85 20

應爲STA $20 85 2C

第十二期P.11，右邊欄列表由底數上第五行

BCF4— 20 8D B7

應改爲BCF4— 20 01 F6

各期程式磁碟磁帶總目錄

1
1 突破傳統的鍵盤
2 捉球遊戲
3 勇闖電腦室
4 迷宮救美
5 將 APPLESOFT 變爲 FORTRAN
6 上、下、左、右移動文字畫面
7 測試工具— BOOLEAN 程式
8 騰空更多磁碟空間 一搬移HELLO
9 重現刪除之檔案
10 電腦時代月曆
11 COMPUTEXT 文字處理
12 時日相距計算程式
13 增強英文字滙能力
14 幼童數學四則運算練習
15 過四關

2
1 貨櫃處理
2 沙蟹幻方局
3 巧破密令
4 美術字繪劃器
5 小階字母處理
6 程式轉載器
7 彩色電視／監視器 色彩測試
8 保留遊戲積分記錄
9 數學套件程式
10 磁碟修護工具
11 加送電腦時代月曆 圖像一個

3
1 DOS 的自動同步追踪器
2 動量不滅定律模擬法
3 自製磁碟索引表封套
4 蘋果編曲器
5 幼童認數字初階
6 賽跑遊戲
7 步步追踪
8 DOS KILLER
9 圍棋遊戲
10 巴哈的古典樂章名作
11 旋轉螢幕畫面
12 雜誌目錄分類歸檔系統
13 神經病醫生一神經鬆弛體操

4
(1) DOS 操作指令簡化 器
(2) TEXT FILE 有些甚麽內容？
(3) 立體迷宮
(4) 英文串字練習
(5) 如何以機械語言來修補 DOS
(6) 電腦動畫製作原理。
(7) 豐富設計的學前兒童 教育程式
(8) 應用數學套件程式(二)
(9) DOS WRITER
(10) 穿梭機哥倫比亞號機
(11) 找回失落了的 STEP 及 TRACE
(12) TRACER控制常式
(13) 將 RAM 卡作爲磁碟機使用
(14) 黑房計時器
(15) 太陽系模擬系統

9
(1) 磁碟檔案黃頁系統。
(2) 增強鍵盤輸入及螢幕輸出功能的系統
(3) 將工具程式按放入 DOS 佔用的三條磁軌上
(4) 快速尋找及磁碟矩陣儲存與提取
(5) 新 DOS 命令改換程式
(6) 任意設定目錄索引顯示模式
(7) 捉迷藏
(8) 古堡之謎
(9) 高空擲物
(10) 有趣的排序
(11) 捉蟲問題及答案
(12) 第八期捉蟲問題答案
(13) 印出高屏像數目字

5
1 私人約會備忘日曆
2 三度空間顯像系統
3 將 CP／M 檔案轉爲 DOS
4 蘋果萬花筒
5 高解像度文字產生器
6 修訂 DOS 操作指令簡化器
7 濃縮圖片儲存空間
8 幼童數學練習—— 數字咭
9 加數機大登場
10 備忘 MEMO紙
11 光速計數器
12 剩餘磁碟空 SECTOR數目
13 16進制輸入數字鍵盤
14 問題棋盤
15 神偷智破夾萬
16 隔山打牛
17 大空船降落任務

6
(1) 私人會計系統。
(2) APPLESOFT 擴展版本。
(3) GO]〔 HELLO 的程式
(4) APPLESOFT 偵錯系統。
(5) DOS 指令再修訂版本。
(6) 布倫時鐘。
(7) AMPERSAND 數基轉換器
(8) 完整索引錄。
(9) 砌積木。
(10) 神奇數字方塊。
(11) 亡命飛車。
(12) 迷宮搜秘。
(13) 太空怪侵略戰。
(14) 累加遊戲。
(15) 爲磁碟命名。
(16) 任意定義鍵盤。
(17) 捉蟲訓練問題。

7
(1) 會說話的蘋果
(2) 私人財政預算系統
(3) 私人收入會計系統
(4) 高解像字符繪劃器
(5) 圖像轉載器
(6) 螢幕文本編輯處理器
(7) 濃縮圖片程式修訂加強版本
(8) 蘋果杯大賽車
(9) 十六個難度的食鬼遊戲
(10) 高爾夫球遊戲。
(11) 空戰驚魂
(12) 美麗圖案設計家
(13) 更多色彩
(14) 列印的困擾
(15) 十一條空 SECTOR
(16) 捉蟲訓練
(17) 80字行咭例子程式

8
(一) APPLESOFT 滙編器
(二) 最簡單的拆帶程式
(三) 英文打字練習
(四) 編印單個LABEL
(五) 高解像頁處理工具箱
(六) 簡易快速的記憶移動方法
(七) 眞正要動腦筋的小學四則運算
(八) 七海霸王 (DEPTH CHARGE)
(九) 原子黑洞
(十) 蛇之噩夢
(十一) 收銀機
(十二) 改良的閃動訊息
(十三) 兩欄索引顯示
(十四) 自動三角函數曲綫繪劃系統
(十五) 任意鍵盤 (三個)
(十六) 每期捉蟲問題
(十七) 奇妙的 &

各期程式磁帶

編號：C#1
1、捉球遊戲
2、勇闖電腦室
3、迷宮救美
4、過四關
5、增強英文字滙能力
6、幼童數學四則運算
7、時日相距計算程式

編號：C#2
1、沙蟹幻方局
2、貨櫃處理
3、數學套件程式
4、巧破密令
5、彩色電視／監視器色彩測試

編號：C#3
1、賽跑遊戲
2、步步追踪
3、圍棋遊戲
4、動量不變定律模擬方法
5、幼童認數字初階

編號：C#4
1、豐富設計的學前兒童教育程式
2、應用數學套件程式之二
3、立體迷宮
4、穿梭機哥倫比亞號
5、英文串字練習

編號：C#5
1、加數機大登場
2、神偷智破夾萬
3、隔山打牛
4、太空船降落任務
5、蘋果萬花筒
6、幼童數學練習 ——數字咭

編號：C#6
1、累加遊戲
2、亡命飛車
3、迷宮搜秘
4、太空侵略戰
5、切積木
6、神奇數學方塊

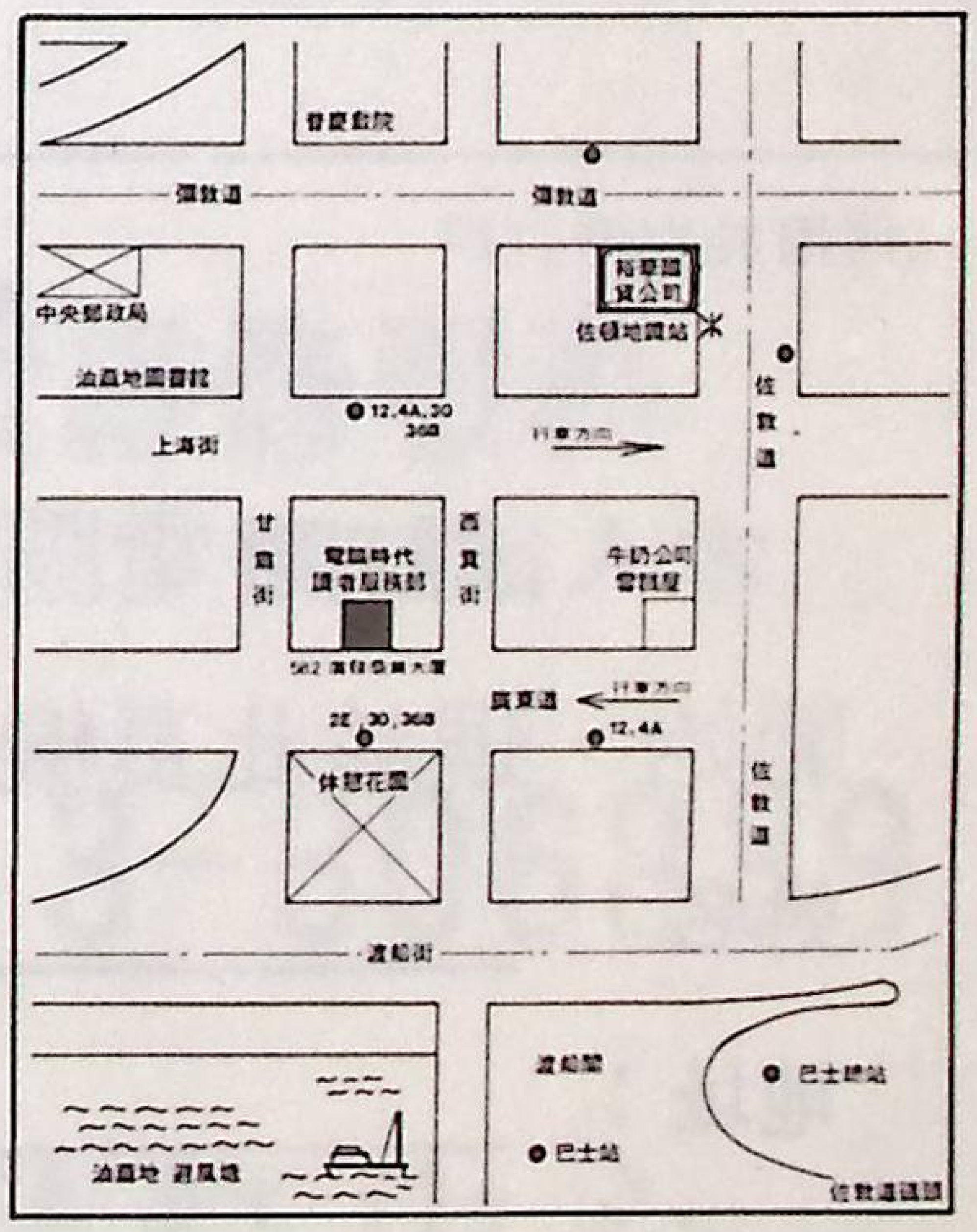
編號：C#7
(一) 十六種難度的食鬼遊戲
(二) 更多色彩
(三) 空戰驚魂
(四) 捉虫問題
(五) 捉虫答案

編號：C#8
(一) 最簡單的拆帶程式
(二) 英文打字練習
(三) 原子黑洞
(四) 蛇之噩夢
(五) 收銀機
(六) 任意鍵盤

精選程式磁碟

編號：CA00A
內容：遊戲及教育程式
遊戲程式
1、捉球遊戲 (第一期)
2、勇闖電腦室 (第一期)
3、迷宮救美 (第一期)
4、過四關 (第一期)
5、沙蟹幻方局 (第二期)
6、貨櫃處理 (第二期)
7、賽跑遊戲 (第三期)
8、步步追踪 (第三期)
9、圍棋遊戲 (第三期)
10、立體迷宮 (第四期)
11、穿梭機哥倫比亞號 (第四期)
教育程式
12、增強英文匯能力 (第一期)
13、幼童數學四則運算 (第一期)
14、數學套件程式 (第二期)
15、動量不滅定律模擬法 (第三期)
16、幼童認數字初階 (第三期)
17、豐富設計的學前兒童教育程式 (第四期)
18、應用數學套件程式之二 (第四期)

編號：CA00B
內容：應用程式精選
1、電腦時代月曆 (第一期)
2、COMPUTEXT 文字處理 (第一期)
3、美術學繪劃器 (第二期)
4、彩色電視／監視器色彩測試 (第二期)
5、保留遊戲積分紀錄 (第三期)
6、自製磁碟索引封套 (第三期)
7、APPLE 編曲器 (第三期)
8、雜誌目錄分類歸檔系統 (第三期)
9、DOS WRITER (第四期)
10、電腦動畫製作原理 (第四期)
11、將 RAM 咭作爲磁碟機使用 (第四期)



歡迎補購各期

程式磁碟磁帶或雜誌

佐敦廣東道582號
廣發商業大廈八樓 三室
TEL：3-312007

電腦時代程式磁碟

每張只售HK$29.90

電腦時代每期都刊登多個精彩的程式，其中必然有你喜歡的。當你讀完文章興緻勃勃的坐下來欲進一步深入了解程式的精彩處時，看見那大堆冗長的列表，就甚麼興趣也沒有了！那麼，你爲何不購買一張「電腦時代程式磁碟」呢！

每期出版的電腦時代所登載的程式，均全部紀錄入一張高品質的磁碟上，並隨雜誌同時出版。爲省卻打入程式的麻煩，徒然浪費你寶貴的時間，免除了花精神捉錯打字的痛苦，今日就請就近的特約經銷處或電腦時代讀者服務部/COMPUHOUSE購備一張。

程式磁碟每張只售港幣 29.90 元。

第13期程式磁碟內容：

*All Programs are in Apple version

(1) 行政思考决策輔助系統——成功人士必備工具
(2) 全能標貼製作器
(3) 大力水手——POPEYE
(4) 火樹銀花賀聖誕——平安夜
(5) 低解像圖輔助繪劃器
(6) 達爾文進化論的電腦模擬實踐
(7) 程式編輯器
(8) 磁碟檔案地圖
(9) 增加DOS的CATALOG暫停功能
(10) 新游標
(11) 40軌磁碟改動器
(12) 餅形圖
(13) 自己設信紙版頭
(14) 捉虫問題及答案
(15) 一行程式得獎作品

本期程式磁碟雙面灌錄　讀者留意

為方便各區讀者購買程式磁碟，我們新增加多個特約經銷處：
以下爲各特約經銷處地址：

中環

百老滙攝影器材有限公司（第五店）
域多利皇后街3號（中環街市側）
電話：5－235513

亞當影音
士丹利街 28 A（陸羽茶樓側）
電話：5－240723

三聯書店讀者服務中心
香港中環域多利皇后街九號（中環街市側）
電話：5~250102

銅鑼灣

百老滙攝影器材有限公司（第十四店）
軒尼斯道529號地下（三越百貨公司對面）
電話：5－770181

太古城

百老滙攝影器材有限公司（第七店）
太古城中心第232店
電話：5－672006

西環

創業雜誌中心
西環屈地街22號地下　明愛中心對面
電話：5-478338

旺角

百老滙攝影器材有限公司　（第1,3,9,10店）
彌敦道758號在下（凱聲戲院側）
電話：3－816931
彌敦道746號地下（麗聲戲院側）
電話：3－943827
豉油街 26 號地下（近商務書局）
電話：3－884331
亞皆老街 60 B 地下（世界書局對面）
電話：3－956367

青年圖書有限公司
亞皆老街 80 號昌明大廈H座1／F
（飛龍快餐店樓上）電話：3－918965

尖沙咀

智源書局
金巴利道 27－33號永利大厦2字樓A座
電話：3 － 678482 3－678414

荔枝角

百老滙攝影器材有限公司（第二店）
美孚新邨第四期百老滙街 79號（影都戲院側）
電話：3－7424923

深水埗

百老滙攝影器材有限公司（第十三店）
青山道148號地下（華聲戲院對面）
電話：3－7295938

觀塘

百老滙攝影器材有限公司（第四、十二店）
物華街3號地下（皇上皇側）
電話：3－419513
同仁街 35 號地下（大家樂側）
電話：3－449940

荃灣

百老滙攝影器材有限公司（第十一店）
沙咀道 230 A（戴麟趾夫人健康院對面）
電話：0－438926

新增設新界屯門特約經銷商：
恒達時電腦公司
屯門華都花園地下商場8號室
電話：0-252577

電腦時代讀者服務部/ CompuHouse

地址：九龍廣東道588號廣發商業大厦八樓三室
（近佐敦道渡船閣）　電話：3-312007

補購各期程式磁碟可前往讀者服務部／COMPU HOUSE

專欄

44 ■ DOS 3.3 專欄—— 40軌磁碟改造器
61 PRODOS縱橫—— PREFIX 命令
62 機械語言入門—— 加減乘除四則運算原理（下篇）

學校電腦輔助教育

64 ■ 達爾文進化論的電腦模擬實踐

兒童電腦輔助教育

66 ■ 低解像圖輔助繪劃器

齊齊樂開心坊

56 ■ 大力水手POPEYE
60 ■ 火樹銀花賀聖誕（音樂程式）

80 最新遊戲軟件評介

古墓之謎
頂烏龜
七海尋金熱

教育軟件評介

85 預備到美國升學的同學適用的兩套軟件

精品店

81 ■ 餅形圖製作器
87 ■ 一行程式參加得獎者
86 ■ 信紙頭設計
52 ■ 新游標
85 ■ 捉蟲訓練

讀者園地

3 COMPUSOFT磁碟月刊廣告介紹
90 優待長期訂戶特惠訂閱優待表格
46 第一至十二期總目錄
21 業餘程式編寫員好消息
89 各期程式磁碟總目錄
53 投稿須知
88 更正

編輯室

編者言 上期刊登了一篇小型的製作「增加RAM 指顯示器」後，讀者反應非常熱烈，並要求我們多刊登同類型的稿件。由今期開始，我們又再增闢一個小型製作專欄，爲大家提高一些簡單而實用的小製作。在此作一個預告：下期我們將教大家改動CONTROLLER咭，令可以在開機時B-OOT動第二部磁碟機。

我們今期嘗試刊登一個很具創意的應用程式〔行政思考決策輔助系統〕。在形式上來說，程式有意模仿第五代電腦——自己會思考推理。希望各位讀者能好好利用。

麥法基先生今期又爲我們寫了一個大力水手的遊戲，遊戲可同時配合MOCKINGBOARD 玩耍。

爲了讓大家對這塊MOCKING BOARD 有更深的認識，我們製作了一個特輯來REVIEW這張咭。

〔40軌磁碟改造器〕是一個値得爲磁碟用家推薦的程式，它能將一張原本是35軌錄有資料的磁碟，在不影响這些資料情況下，再INIT 多5條軌，讓磁碟的空間增加多百分之十五。

我們又爲大家介紹一個爲磁碟加上保護的方法，絕不容錯過今期內容，相信各位必定感到十分滿意。

■ Program saved on disk

特稿

令APPLE發聲趨向完美的MOCKING BOARD 8
以MOCKINGBOARD來製造立體聲音响的技巧 11
REBOOT DOS 45
蘋果中的蟲 82
改動BOOTING PROGRAM FILENAME 83
APPLESOFT與MBASIC之比較 84

高級篇

MAILMERGE的操作備忘 WORDSTAR 高級應用篇之三 39
令VISICALC變爲會思想的工具 40
以標準格式編印FINANCIAL STATEMENT d BASE II 進階篇 42

應用軟件庫

■ 行政思考決策輔助系統—成功人士必備工具 26
■ 全能標貼製作器 17

軟件縱橫談

CAT-GRAPHICS爲BASIC提供55個新指令 48
CP/M 55 個公用程式——POWER（下篇） 53
室內設計軟件包—HI-RES ARCHECTURAL DESIGN 54

小型製作專欄

會發聲的鍵盤 15
爲PAL咭的文字顯示加上顏色 55

蘋果工具箱

■ 程式編輯器——蘋果缺乏的工具 68
■ 磁碟檔案地圖顯示器 71
■ 暫停CATALOG功能指令 79

BACK UP資料館

最具威力的拷貝程式之一 LOCKSMITH 5·0 使用方法 77
程式保護技巧之一 VTOC的移動 75
最快的COPY程式：DIVERSI COPY 79
MURDER ON THE ZINDERNEUF 抄法 88

電腦時代

NO.13

本刊每月廿五日出版

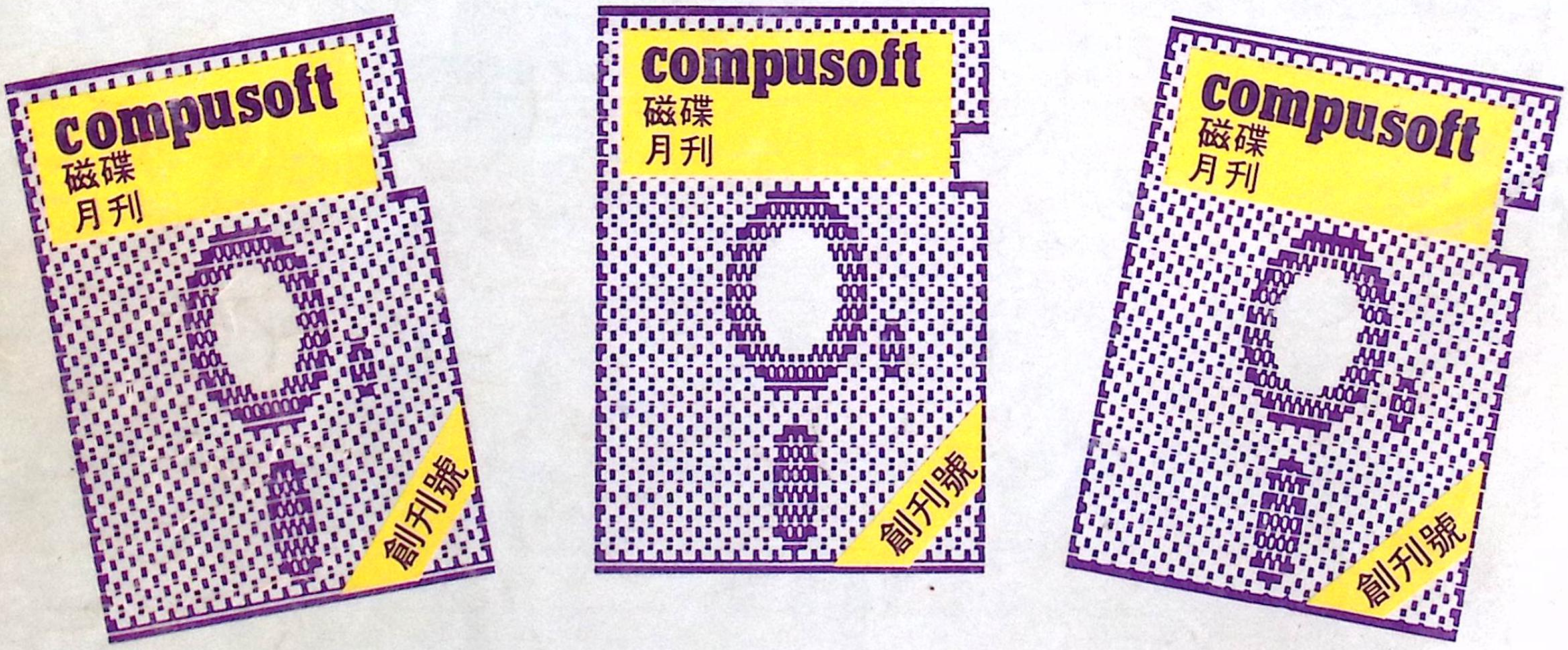
■ 出版 電腦時代出版社 ■ 督印 伍小華 ■ 主編 伍小華 ■ 編輯 電腦時代編輯部 ■
■ 軟件支援 CompuHouse ■ 通訊地址 九龍中央郵箱71193號 ■ 通訊電話 3-312007 ■
■ 讀者服務部 廣東道582號廣發商業大厦703室 ■ 承印 永恒印刷公司 ■ 發行 同德書報社 ■

香港蘋果電腦用家踏入歷史性新里程

一本有聲、有畫、可讀、可玩的雜誌

COMPUSOFT 磁碟月刊

創刊號十一月中隆重出版



電腦時代出版社開創本港出版界的先河，首先出版發行一本創新革命性的雜誌——COMPUSOFT磁碟月刊，爲全港數以萬千計的蘋果電腦用家及有關人士帶來一個重大喜訊。

COMPUSOFT 磁碟月刊（DISK MAGAZINE）是一本內容豐富、題材廣泛的綜合性電腦教育、娛樂及知識性雜誌，無論是電腦發燒友，在職青年、學生、學校教職員、專業人士，甚至乎家中每個成員都適合〔閱讀〕的雜誌。只要閣下擁有一部蘋果或蘋果相容的電腦及一部磁碟機，就可成爲COMPUSOFT磁碟月刊的訂閱戶。

COMPUSOFT磁碟月刊每月出版一次，每期包括兩張雙面灌錄的5$\frac{1}{4}$吋磁碟。

歡迎各公司刊登廣告，推廣期間費用特廉

查詢詳情可電 3-312007　　電腦時代廣告部洽